Die computationale Wende in der Statistik

Wie hat sich die statistische Methodik und ihre Anwendung verändert?

Abschlussarbeit

zur Erlangung des akademischen Grades
Bachelor of Science

vorgelegt von

Christina Ecker

am 27. September 2019

Institut für Statistik
Ludwig-Maximilians-Universität
München

Betreuer: Prof. Dr. Thomas Augustin
PD Dr. Rudolf Seising

Abstract

Das Ziel der vorliegenden Bachelorarbeit war es, die Gründe, den Verlauf und die Ergebnisse der computationalen Wende in der Statistik aufzuzeigen. Der Schwerpunkt der Arbeit lag vor allem darauf, wie sich dadurch die Statistik und ihre Anwendung verändert hat. Dabei wurde zunächst das Hintergrundwissen über die Geschichte moderner Computertechnologien geliefert, um daraufhin zeitlich parallel die Entwicklung der computationalen Wende darzulegen. Es wurden die historischen Hintergründe ausgeführt und dazu die statistischen Methoden zum jeweiligen Forschungsstand beschrieben. Als besonderer Aspekt der computationalen Statistik wurde das R Project hervorgehoben. Nicht nur dessen Entstehungsgeschichte sowie heutige Errungenschaften wurden beschrieben, auch zeigt eine Zitationsanalyse von Publikationen, welche die Programmiersprache R zitieren, die Phase der steigenden Akzeptanz von R und der Programmierumgebung auf, das stetige Wachstum ihrer Paketerweiterungen bis hin zur Anwendung der Programmiersprache in verschiedenen Wissenschaftsbereichen und Ländern bis heute.

Inhaltsverzeichnis

1 Einleitung

> „There is some sort of law working here whereby statistical methodology always expands to strain the current limits of computation.“

(Efr00)

Auch wenn Bradley Efron dieses Zitat schon vor knapp 20 Jahren niederschrieb, gilt dieses „Gesetz“ noch heute. Denn als ich es bei der Recherche zu dieser Arbeit entdeckte wurde mir sofort klar, dass es meine Fragen sowie meine Ergebnisse in nur einem Satz zusammenfasst. Seit Beginn meines Statistik Studiums stelle ich mir die Frage, seit wann die Statistik schon so betrieben wird wie sie mir an der Universität gelehrt wird. Denn zu den theoretischen und statistischen Grundlagen und Verfahren haben sich in meinem Studium auch Grundkenntnisse der Programmierung sowie statistische Software gesellt. Vor allem die Programmiersprache R ist Gegenstand fast jeder Vorlesung oder Übung. Auch in Praxisprojekten wird der sichere Umgang mit der Programmiersprache erlernt. Doch seit wann ist die Statistik eben diese Mischung aus statistischen und informatischen Elementen? Wie kam es dazu, dass die Statistik nicht mehr nur auf dem Blatt und per Hand, sondern durch Programmcodes und Algorithmen praktiziert wird? Und vor allem, wie haben diese Möglichkeiten die statistische Methodik und somit auch ihre Anwendung verändert?

In dieser Abschlussarbeit habe ich mich mit diesen Fragen beschäftigt und die genauen Gegebenheiten der computationalen Wende in der Statistik recherchiert und analysiert. Dabei eröffneten sich mir interessante Einblicke in die geschichtliche Entwicklung der Statistik. Zum einen, wie durch eine Idee beim Kartenspiel zu Zeiten des zweiten Weltkriegs eine Revolution gestartet wurde, die nicht nur die Statistik geprägt hat. Zum anderen die Vision zweier Programmierer, die es ermöglichten, dass ich als Studentin die statistische Programmierung erlernen und nutzen kann, ohne dabei kostenintensive Lizenzen erwerben zu müssen. Dies begründete, verbunden mit der rasanten Entwicklung des Computers, spannende Forschungsfelder wie die Künstliche Intelligenz und das daraus resultierende Machine Learning.

Dabei werde ich zeigen wie der technische Fortschritt und der Wunsch rechenintensive, statistische Methoden zu erleichtern zusammenhängen und wie viele weitere Zweige der Wissenschaft davon profitieren. In meiner Analyse zeige ich auf wie das R Project zu einem Beispiel und auch vielleicht einem späteren Wegbereiter der computationalen Statistik wurde, indem ich eben dieses benutze, um dessen steigende Verbreitung in den verschiedenen Forschungsgebieten und Ländern der Welt aufzuzeigen. Dafür ist diese Arbeit in zwei Teile gegliedert. Zunächst wird in Kapitel 2 der historische Hintergrund der Entwicklung moderner Computertechnologien bereitgestellt, als Vorkenntnisse für die wissenschaftshistorische Entwicklung der computationalen Statistik des dritten Kapitels. Beginnend in den 1940er Jahren des zweiten Weltkriegs und den Forschungen des Manhattan Projekts über

die Sampling-Methoden der 80er und 90er Jahre, werden auch die Entwicklung der Programmiersprachen bis hin zur Gründung des R Projects und dessen heutigen Wirken aufgezeigt. Auch die Theorien der Künstlichen Intelligenz und ihrer Teilbereiche, dem Machine-Learning und Deep-Learning sowie die daran entstandenen Diskussionen werden im ersten Teil der Arbeit dargelegt. Der zweite Teil beschäftigt sich im vierten Kapitel mit der Zitationsanalyse der Programmiersprache und -umgebung R, um die theoretischen Ergebnisse des ersten Teils noch weiter aufzuzeigen. Es analysiert die Verwendung der statistischen Programmiersprache in veröffentlichten Publikationen. Auch wird die Aktualität der Programmierumgebung R und deren Anpassung an die statistischen Methoden gezeigt.

2 Hintergrund: Meilensteine der modernen Computertechnologien

Dieses Kapitel gibt Aufschluss darüber, wie sich der Computer zur Zeit der computationalen Wende in der Statistik entwickelt hat. Dabei werden die Meilensteine behandelt, die für diese Entwicklung am wichtigsten erscheinen. Es zeigt die Geschwindigkeit des technischen Fortschritts und die dadurch bereitgestellten Mittel und Möglichkeiten für die Statistik auf. Auch wird es eine kurze Einführung mit Erklärungen darstellen, um ein Grundwissen zu bieten welches für das nächste, zeitlich parallele, Kapitel 3 benötigt wird.

Zunächst jedoch werden einige Begriffe definiert, die im folgenden Kapitel verwendet werden, siehe z.B. (Bru18a) und (Bru18b). Der Begriff der Rechentechnik, bedeutet unter Anderem das Vorgehen beim manuellen oder maschinellen Rechnen, hier in Rechenmaschinen oder Rechnern. Ein Rechner ist ein eben solches Rechengerät, später auch Computer genannt. Dieser Rechner kann mechanischer Natur sein, also durch mechanische Betätigung betrieben oder auch elektrisch, sprich durch elektrischen Strom und Schaltungen betrieben oder später elektromechanisch, das heißt eine Mischung beider Vorgänge. Dabei bedeutet die Entwicklung vom Dezimal- zum Binärrechner eine Umstellung von Berechnungen über das dezimale Zahlensystem und die Ziffern 0 bis 9, auf das binäre beziehungsweise das duale System, eine Darstellung von nur zwei Ziffern, üblicherweise 0 und 1. Den Rechner selbst bezeichnet man hier als Hardware, also die Geräte und Bauteile. Im Gegensatz dazu meint die Software Programme und Daten. Software ist stets an Hardware gebunden. Das Wort Programmieren hat sich mit dem technischen Fortschritt gewandelt. Früher programmierte man Rechenmaschinen über Steckverbindungen, wobei man später damit die Software steuerte. Auch der Begriff der Informatik wandelte sich. Zunächst mit dem Schwerpunkt Hardware, beschäftigte sich die Informatik immer mehr mit der Software und der Programmierung. Ebenfalls zu erwähnen ist die Steuerung, welche sich von der Ablaufsteuerung, also einem schrittweisen Ablaufen von Befehlen hin zum Prozessrechner entwickelt hat, der Datenverarbeitung in Echtzeit. Auch der Speicher des Rechners hat sich verändert. Als Funktionseinheit innerhalb eines Computers welche Daten aufnimmt, aufbewahrt und wieder abgibt, hat sich diese von einer Lochkarte zu Magnetbändern oder -platten ausgebaut: Die Speichersteuerung von der externen Lochkartensteuerung zur internen Programmsteuerung, der sogenannten Speicherprogrammierung.

Zuletzt muss der Begriff des Computers erläutert werden. Ursprünglich bezeichnete dieser einen rechnenden Menschen. Dann erst stecktafelgesteuerte, programmgespeicherte und schließlich speicherprogrammierte Rechenmaschinen. Die Bezeichnung ist also auf den jeweiligen Stand der Technik angepasst, wird aber in verschiedenen Quellen oft unterschiedlich ausgelegt. In dieser Arbeit wird der Begriff des Computers aber im weiteren Sinne verwendet.

2.1 Vor dem zweiten Weltkrieg: Von der Lochkarte zur Turingmaschine

Am Ende des 19. Jahrhunderts war das Mittel zur Datenverarbeitung, -speicherung und -bereitstellung die Lochkarte. Die Lochkartenmaschinen wurden besonders für die Statistik, welche damals noch einen verwaltungstechnischen Hintergrund hatte, oder die Buchhaltung genutzt, siehe z.B. (Bru18a). Erfinder der Lochkartenmaschine und der zusätzlich benötigten Tabelliermaschine zum Auswerten und Sortieren der Lochkarten, war Herman Hollerith. Dieser entwickelte sie primär für den Einsatz in Volkszählungen vor und besonders während des zweiten Weltkriegs. Sein erster Einsatz fand bei der 11. amerikanischen Volkszählung 1890 statt, siehe z.B. (Bru18b). Das Bedürfnis, relativ aufwendige mathematische Berechnungen und Vorgänge zu automatisieren begründete den heutigen digitalen Wandel. Zunächst von der Mechanik zur Elektronik, dann von der Analog- zur Digitaltechnik. Die erste Wende setzte sich Mitte des 20. Jahrhunderts durch. In Hinblick auf den Krieg war die Nachfrage nach Rechenleistung groß. Die Ermittlung von Flugbahnen der Geschosse, das Entziffern von feindlichen Funksprüchen und die Forschung am Bau der Atombombe, oft im Auftrag von US-amerikanischen Hochschulen, dem Militär oder britischen Forschungseinrichtungen waren die Wegbereiter für immer schnellere Rechenmaschinen, siehe z.B. (Bru18a) und (Bru18b). Auch in Deutschland begann der deutsche Konrad Zuse bereits 1936 mit dem Bau eines der ersten Rechner, dem Z1. Dieser hatte noch mechanische Schaltglieder inne und war so groß wie ein Doppelbett, siehe z.B. (Les10). Damit wollte er sich eine Maschine zur Erleichterung bei der statistischen Berechnung bauen, siehe z.B. (Bru18b). Auch in den 1930er Jahren begannen unabhängig voneinander die Amerikaner Howard Aiken, George Stibitz und John Atanasoff digitale Relais- und Röhrenrechner zu bauen und der britische Alan Turing begann mit den Plänen einer universellen Machine, der heute bekannten Turingmaschine. Diese stellte er dann 1936 in seiner Arbeit „On computable numbers, with an application to the Entscheidungsproblem“ dar (Tur36). Das Konzept der Turingmaschine ist nicht eine echte und funktionstüchtige Maschine zu bauen, sondern stellt ein Modell dar, um Erkenntnisse und Fähigkeiten über die Rechenmaschinen zu erlernen und weiterzuentwickeln. Denn es ist eine Maschine mit der man jedes beliebige Rechenproblem lösen kann, das auch durch einen Mensch lösbar ist, nur eben wesentlich schneller. So bezeichnet bis heute der Befriff ‚turingmächtig‘ oder ‚turingvollständig‘ eine Datenverarbeitungsmaschine die universell programmierbar ist, siehe z.B. (Bru18b). Ein solches theoretisches Modell war und stellt für die Informatik sowie für die Entwicklung von Programmiersprachen und der Automatentheorie ein lehrreiches Modell dar.

Das absichtlich recht einfache und leicht zu analysierende Konstrukt der Turingmaschine ist vereinfacht in Abbildung 2.1 zu sehen. Dabei ist ein unendlich langes Band in einzelne Felder aufgeteilt in denen jeweils ein binärer Zahlenwert gespeichert ist, zum Beispiel hier entweder 0 oder 1. Eine Maschine, bestehend aus einem Programm und einem Lese- und Schreibkopf, schreitet das Band entlang und liest die Zahlen aus. Die Maschine kann verschiedene innere Zustände annehmen und hält auch in einer Tabelle fest was passiert ist, also welcher Zustand bei welcher Zahl. Die Maschine kann ihren Zustand aufgrund von festgelegten Regeln ändern und auch die Zahlen auf dem Band überschreiben. Jetzt bringt

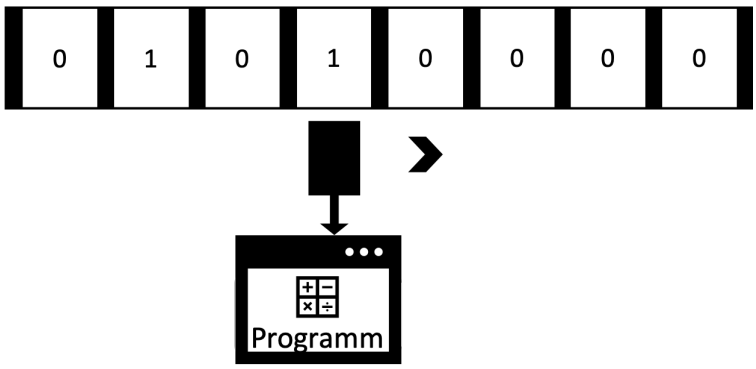

Abbildung 2.1: Vereinfachte Darstellung einer Ein-Band-Turingmaschine: Das Programm bewegt den Lese- und Schreibkopf am Band, Eigene Darstellung

man der Maschine zunächst zwei Zustände bei. Zustand A ist die Zahl auf dem Band von 0 auf 1 oder 1 auf 0 zu überschreiben, den inneren Zustand auf B zu schalten und auf dem Band ein Feld nach rechts zu rücken. Zustand B ist die Zahl auf dem Band beizubehalten, den inneren Zustand auf A zu stellen und wieder auf dem Band nach rechts zu rücken. Zu Anfang bestehen die Felder auf dem Band ausschließlich aus Nullen. Startet man nun das Programm, beginnt die Maschine auf dem Band Schritt für Schritt nach rechts zu wandern und abwechselnd 0 und 1 zu hinterlassen bis es ein Stopp-Kriterium erhält. Gestaltet man nun die programmierten Regeln beliebig kompliziert, mit vielen inneren Zuständen, dann hat man eine universell programmierbare Rechenmaschine. So wird auch in Robert Irving Soares Buch: „Turing computability“ gesagt: „Turing's achievement was to determine not what machines could compute, but what human beings could compute with enough resources of time and space for the computation.” (Soa16).Die Turingmaschine zeigt also auf, zu welchen Berechnungen ein Mensch fähig wäre, würden ihm unendlich Zeit und Raum zur Verfügung stehen.

2.2 Der Computer als Waffe im zweiten Weltkrieg

Zu Beginn der 1940er Jahre fiel der Startschuss zur rechnergestützten Statistik. Dies stellt auch die erste Phase der Digitalisierung dar. Der heutige Digitalrechner hatte, wie schon erwähnt seine Ursprünge in verschiedenen Ländern: Deutschland, England und den USA. In Deutschland war es das Werk eines einzelnen Forschers. Ansonsten waren Hochschulen und Forschungseinrichtungen sowie die Industrie dahinter. So kann man keinen eindeutigen Erfinder des Computers festmachen. Aufgrund des zweiten Weltkriegs von 1939 bis 1945 waren die Länder und Entwickler weitgehend voneinander abgeschnitten. Auch war eine Zusammenarbeit meist nicht im Interesse der einzelnen Beteiligten, denn vor allem die Briten arbeiteten unter großer Geheimhaltung an einer Maschine zur Entschlüsselung feindlicher Funksprüche. So kann man sagen, dass es viele „Geburtshelfer“ des Computers gab (Bru18b).

John von Neumann, welcher in Abschnitt 3.1 noch eine große Rolle für die computationale

Statistik spielen wird, war derzeit an dem Bau der Atombombe beteiligt. 1945 schrieb er eine Arbeit über speicherprogrammierte Maschinen und erfand so die Von-Neumann-Architektur, beziehungsweise den Von-Neumann-Rechner. Damit war es nun möglich eine Änderung am Programm eines Computers vorzunehmen, ohne etwas an der Hardware zu verändern. Zuvor entwickelte Rechner waren an ein festes Programm gebunden. Somit erarbeitete er dabei eine Turingmaschine, jedoch speicherprogrammiert, in Form eines Modellkonzeptes eines Computers mit einem gemeinsamen Speicher für einerseits Programmbefehle und andererseits die Daten. Dies gilt bis heute noch als Grundlage eines jeden Computers, siehe z.B. (Bru18b). Konrad Zuse merkte auch, dass diese Art der Programmsteuerung für vielfältige Zwecke einsetzbar war und entwickelte den Z3, die erste funktionstüchtige, programmgesteuerte binäre Rechenmaschine.

1945 stellte Zuse seinen Z4 als Nachfolger des, durch eine Bombe zerstörten, Z3 vor. Außerdem präsentierte die Firma IBM seinen ersten Großrechner, den Automatic Sequence Controlled Calculator (ASCC), welcher 15 Meter lang und fünf Tonnen schwer war, siehe z.B. (dpa11). Zudem entwickelten sie 1946 bereits einen ersten kommerziellen elektronischen Rechner: den IBM 603. Nach drei Jahren Bauzeit erschien der Electronic Numerical Integrator and Computer (ENIAC). John Eckert und John Mauchly entwickelten den ersten vollelektronischen digitalen Universalrechner. Zunächst für das US-Heer als Spezialmaschine für ballistische Berechnungen gedacht, erkannten sie schnell das Potential dieses leistungsfähigen Rechners. Weitere Nachfolger waren der Universal Automatic Computer (UNIVAC) und Mathematical Analyzer Numerical Integrator And Computer Model (MANIAC), siehe z.B. (Bru18a). In einem Bericht von 1947 über den ENIAC heißt es: „Dieser Computer kann die Flugbahn eines Geschosses, die in zweieinhalb Sekunden durchlaufen wird, in eineinhalb Sekunden berechnen. Die Programmierung dauert eineinhalb Tage.“, nach (Tod18). Trotz des hohen Programmieraufwands war dies eine gewaltige Entwicklung für den Computer.

Auch Alan Turing ist ein Verdienst im zweiten Weltkrieg zuzuschreiben. So hat er zusammen mit einem Team in Bletchley Park eine Maschine gebaut, welche die verschlüsselten Funksprüche der deutschen Schlüsselmaschine ENIGMA wieder entzifferte, um so einen Vorsprung gegenüber den feindlichen Plänen zu erlangen. Diese Maschine wird auch die Turing-Bombe genannt, siehe z.B. (Bru18b).

2.3 Die Entwicklung ab 1950: Innovation in Serie

„There is no reason for any individual to have a computer in their home.“

Aus heutiger Sicht erscheint dieses Zitat von Informatiker Ken Olsen nach (SdKS03) scherzhaft, doch trotz der startenden Serienproduktion kommerzieller Computer Anfang der 1950er Jahre, waren diese noch nicht erschwinglich für die breite Bevölkerung. 1951 wurde der Darmstädter Elektronischer Rechenautomat (DERA) und der bereits erwähnte UNIVAC 1 entwickelt. 1952 gelang es IBM mit ihrem Magnetbandspeicher IBM 726 die Datenmenge von 35.000 Lochkarten zu speichern, siehe z.B. (dpa11). Ein weiterer Meilenstein von IBM stellt 1954 eine Übersetzung vom Russischen ins Deutsche durch einen weiteren hauseigenen

Computer dar. 1953 stellte Zuse dann seinen Z5 vor, bei dem es sich jedoch noch um eine Einzelanfertigung handelte. Erst die Relaismaschine Z11 stellte das erste seriell gefertigte und programmgesteuerte Rechengerät Deutschlands dar. Es wurde für technische und wissenschaftliche Anwendungen programmiert, siehe z.B. (Bru18b). Auch im Hinblick auf die Software wurden in den 1950er Jahren Fortschritte gemacht. Formula Translation (FORTRAN), die erste hohe Programmiersprache wurde 1954 von dem IBM-Programmierer John Backus entwickelt. Sie sollte besonders für numerische Berechnungen in der Wissenschaft, Technik und Forschung angewendet werden, siehe auch Abschnitt 3.3. 1955 bauten die Bell Laboratories den Transistorized Airborne Digital Computer (TRADIC), 1956 entwickelte IBM das erste Plattenlaufwerk und der UNIVAC 2 kam auf den Markt. Neben den bereits erwähnten Unternehmen reihte sich auch Siemens mit seinem Siemens 2002 in die serielle Produktion ein, siehe z.B. (Bru18b).

Auch in den 1960er Jahren lag das Hauptaugenmerk auf der Entwicklung immer kleinerer, kostengünstigerer und leistungsfähigerer Rechner. Jedoch waren die Großrechner für kleinere und mittlere Unternehmen weiterhin zu teuer. Ende des Jahrzehnts jedoch stieg das Unternehmen Nixdorf Computer AG in den Markt ein und schloss diese Marktlücke. Mit dem Nixdorf 820, dem ersten Tischrechner, wurde der Computer direkt am Arbeitsplatz ermöglicht. Durch die erschwingliche Anschaffung profitierten immer mehr Kleinunternehmen und stellten auf die EDV, die elektronische Datenverarbeitung, um, siehe z.B. (Igg18). 1968 wurde bereits die erste Computermaus vorgestellt, jedoch fand diese erst einige Jahre später ihren Einsatz, als grafische Benutzeroberflächen eingeführt wurden. Nach der Einführung des Computers in immer mehr Lebensbereiche stellt das Internet die zweite Welle der ersten Phase der Digitalisierung dar. Dies wird auch als die "Vernetzung"bezeichnet. 1969 wurden bereits die ersten Computer per Internet verbunden, siehe z.B. (Bru18a).

Die nächste große Innovation war zu Beginn der 70er Jahre der Mikroprozessor durch die Firma Texas Instruments. Dies ist ein Prozessor in sehr kleinem Maßstab basierend auf einem Mikrochip. Somit wurden die Computer immer kleiner und leistungsstärker. 1971 wurde, unter Beteiligung von IBM, die Diskette entwickelt, siehe z.B. (dpa11). Die Forschung machte rasante Fortschritte. Neben den immer mehr verbauten Mikroprozessoren wurden ab 1972 Mehrprozessorsysteme verwendet, um die Leistungsfähigkeit zu steigern. 1973 erschien der Xerox Alto, der erste Computer mit Graphical User Interface (GUI), also einer grafischen Benutzeroberfläche sowie der dazu benötigten Computermaus. Dies ebnete den Weg für die ersten Personal Computer, kurz PC. Diese Computer waren nun in Größe und Leistung für den täglichen Gebrauch konzipiert und sollten für jeden Laien bedienbar sein. Denn bisher war die Computernutzung auf Wissenschaftler oder Techniker beschränkt. Erste PCs stellten 1976 der Apple 1 von Apple Computer und der Commodore PET, kurz Personal Electronic Transactor, von Commodore Internationals dar. IBM trug seinen Computer mit der Serienbezeichnung 5100 bei, den ersten tragbaren Computer, und gab somit den Startschuss für den heute bekannten Laptop. Noch 1949 schrieb die britische Zeitung „The Sun“ nach (Pro16): „The future? The „brain“ [computer] may one day come down to our level and help with our income-tax and book-keeping calculations. But this is speculation and there is no sign of it so far.”. Sie hat wohl nicht damit gerechnet, dass schon nach weniger als drei Jahrzenten jeder einen Computer für genau diese Zwecke zuhause hat.

Die 1980er Jahre waren vom Durchbruch des Heimcomputers geprägt. Damit ist ein PC gemeint, der primär für die Unterhaltung vorgesehen war. Da er sehr erschwinglich war, kamen immer mehr Bevölkerungsschichten in den Genuss eines Heimcomputers. Dabei war die Tastatur über diese man den PC über Kommandozeilenbefehle bediente in das Gehäuse integriert. Der 1982 auf dem Markt erschienene Commodore C64 gilt dabei als der am meisten verkaufte Heimcomputer der 80er Jahre. Vor allem seine besonders gute Eignung für Computerspiele verschaffte ihm eine große Fangemeinde, siehe z.B. (Kon). Doch auch andere Unternehmen erreichten wichtige Meilensteine. 1984 besticht der Apple Macintosh durch seine besonders benutzerfreundliche Oberfläche. Im November 1985 kommt Microsofts erstes Betriebssystem Windows 1.0 auf den Markt und Commodore veröffentlicht den ersten Multimediacomputer, genannt Amiga.

Und so brachten die 1990er Jahre die dritte Welle der ersten Phase der Digitalisierung, das World Wide Web (WWW). Das Europäische Labor für Teilchenphysik Cern in Genf erfand diesen Dienst unter der Leitung des Physikers Tim Berners-Lee, siehe z.B. (Bru18a). Dabei muss das Internet vom WWW unterschieden werden, auch wenn es oft synonym verwendet wird. Denn das WWW nutzt das Internet als weltumspannendes Netz aus vielen einzelnen Computernetzwerken als Dienst zur Datenübertragung von Webseiten. Dafür wird ein Browser benötigt, siehe z.B. (foc09). Die Entwicklung des WWW wurde auf drei Säulen begründet, zum einen die Hypertext Markup Language (HTML) mit welcher beschrieben wird, wie Webseiten durch Links formatiert werden. Weiter das Hypertext Transfer Protocol (HTTP), das die Sprache definiert, den der Computer zur Kommunikation über das Internet verwendet. Und zuletzt der Universal Resource Identifier (URI), mit welchem Dokumentenadressen erstellt und aufgerufen werden, siehe z.B. (foc09). Im weiteren Verlauf der 1990er Jahre wurden immer bessere Prozessoren entwickelt und die Betriebssysteme ständig verbessert und angepasst. Zum Beispiel wird die Linux Version 1.0, wie auch in Abschnitt 3.3 erwähnt, freigegeben und Windows 95 kommt auf den Markt, siehe z.B. (Kon).

2.4 Das digitale Zeitalter

Der Beginn des 21. Jahrhunderts leitete die zweite Phase der Digitalisierung ein. Der Computer ist im privaten wie beruflichen Alltag allgegenwertig. Und nicht nur dort. Mit der Einführung des internetfähigen Mobiltelefons, im Laufe der Zeit auch Smartphone genannt, war der Computer nun auch stets in der Hosentasche dabei, siehe z.B. (Bru18a). 2008 wurde zum Beispiel Googles Smartphone-Betriebssystem Android in Betrieb genommen, siehe z.B. (Kon). Diese Phase der Digitalisierung ist vor allem geprägt von den Kommunikations- und Vernetzungsmöglichkeiten der Benutzer. E-Mail-Verkehr revolutioniert nicht nur die Alltagskommunikation, sondern auch die Geschäftswelt. Besonders die am Anfang der 2000er Jahre aufkommenden Sozialen Netzwerke vernetzen nun Anwender auf der ganzen Welt, siehe z.B. (Ste15). Nicht nur, dass viele Haushalte bereits mehrere Computer besitzen, stationär sowie als Laptop, beginnt 2010 durch Apples iPad der Trend des Tablet-Computers, siehe z.B. (Kon). Wie in Kapitel 3.4 genauer nachzulesen ist, werden Computer nun auch beispielsweise für Bildbearbeitung genutzt und der unbegrenzte Zugang zu Wissen mithilfe von Suchmaschinen und Datenbanken ist stets gegeben.

Ein heutiges Smartphone hat die 120 millionen-fache Rechenleistung des Computers der Apollo Mondmission der Nasa von 1961 bis 1972 und die Leistung eines mittlerweile veralteten Apple iPad 2 entspräche 1994 der eines Supercomputers, um nur zwei Beispiele der rasanten technischen Entwicklung aufzuzeigen, siehe z.B. (Gru16). Denn diese lässt sich sogar durch eine Gesetzmäßigkeit erklären. Gordon Moore, ein Mitbegründer des Halbleiterherstellers Intel, formulierte 1965 in seinem Artikel: „Cramming more components onto integrated circuits", was später als das Mooresche Gesetz bekannt wurde, siehe (Moo65). Demnach soll sich die Anzahl der Schaltkreiskomponenten auf einem Computerchip jedes Jahr verdoppeln, z.B. nach (Gru16) und (Kli19). Später korrigierte er seine Aussage auf eine Verdoppelung alle zwei Jahre. Dies geht auch mit einer verdoppelten Leistung einher und wurde somit zum Leitsatz für Innovationen. Diese vermeintliche Gesetzmäßigkeit konnte bisher eingehalten werden, was nicht zuletzt ein Grund für die rasante Entwicklung des Computers ist. Auch wenn einige Quellen ein Ende des Mooreschen Gesetzes vermuten, siehe z.B. (Gru16) und (Kli19), ist es sicherlich nur ein weiterer Anreiz für die Forschung noch bessere und schnellere Innovationen im Bereich der Computertechnologien zu entwickeln.

3 Ausgewählte Aspekte der geschichtlichen Entwicklung in der computationalen Statistik

In diesem Kapitel werden nun, basierend auf den Kenntnissen der Computertechnik des vorherigen Kapitels, Highlights dargelegt, welche maßgeblich zur Entwicklung der computationalen Statistik beigetragen haben. Zuerst werden die Markov-Chain und Monte-Carlo Methoden genannt, gefolgt von Sampling und Resampling Techniken. Daraufhin wird das R Project von seiner Entstehung bis hin zu dessen Rolle für die Statistik erläutert. Als vierter Aspekt der Entwicklung wird die Methodik der Künstlichen Intelligenz mit seinen Teilbereichen des Machine Learning sowie Deep Learning aufgezeigt.

3.1 Das Manhattan Projekt: Markov-Chain-Monte-Carlo

Um die Entwicklung der computationalen Wende in der Statistik zu beschreiben ist es unabdingbar die Markov-Chain-Monte-Carlo (MCMC) Methoden als erstes zu erwähnen. Somit startet die Entwicklung hin zur rechnergestützten Statistik in den 1940er Jahren mit dem auf den ersten Blick fachfremden Manhattan Projekt und bahnte so den Weg für die, um es mit den Worten von Mitchell Watnik, Professor der California State University zu beschreiben, „Renaissance der statistischen Modellierung“ (Wat11). Das Manhattan Projekt, welches auch unter der Tarnbezeichnung „Manhattan Engineer District“ operierte, war ein militärisches Forschungsprojekt der Vereinigten Staaten von Amerika zur Entwicklung und dem Bau der Atombombe im zweiten Weltkrieg. Geleitet wurde dieses von Physiker J. Robert Oppenheimer und beherbergte 150.000 Wissenschaftler und Mitarbeiter, siehe z.B. (Neu51). Den Mathematikern John von Neumann und Stanislaw Ulam gelang es im Zuge dieses Projekts computergestützt Zufallszahlen zu generieren, die entscheidende Basis für die Anwendung der Monte-Carlo Methoden. John von Neumann gilt bis heute als einer der Begründer der Informatik. Seine Hauptaufgabe im Manhattan Projekt war die Untersuchung des Überschallflugs und der Plutoniumbombe, sowie Verfahren zur Lösung von bestimmten Differentialgleichungen. Ulam war schon früh an den Entwicklungen von Nuklearwaffen beteiligt und arbeitete an theoretischen Zusammenhängen im Manhattan Projekt. Er ist unter anderem für den ‚Satz von Ulam‘ bekannt, einem Lehrsatz der Maßtheorie zu Eigenschaften von Borelmaßen.

Durch die Überlegungen von John von Neumann und Stanislaw Ulam bei einem Solitär Spiel wurde der Weg der Monte-Carlo Methode gebahnt (Wat11). Die Erkenntnis, dass sich

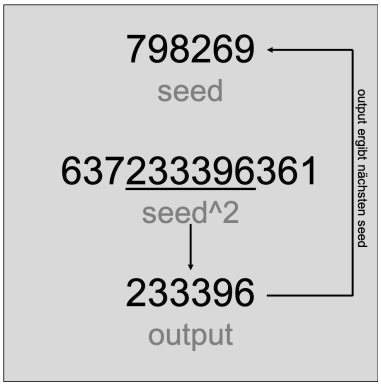

Abbildung 3.1: Ein Beispiel zur Mittelquadratmethode, Eigene Darstellung

die Gewinnwahrscheinlichkeit approximativ darstellen lässt, wenn man das Spiel nur genügend oft simuliert, entwickelte sich zu einer Möglichkeit schwer bestimmbare Integrale durch stochastische oder randomisierte Algorithmen zu approximieren (Wat11). So sagte von Neumann selbst auf einer Konferenz im Jahr 1949: „We see then that we could build a physical instrument to feed random digits directly into a high-speed computing machine and could have the control call for these numbers as needed.“ (MGNR12). Sie haben also entdeckt, dass sie bei Bedarf Zufallszahlen mit Hilfe des Computers erhalten konnten, verstanden aber schnell, dass diese Zahlen einem echten Zufall entsprechen. Diese kann man nicht reproduzieren, was für wissenschaftliches Arbeiten aber unabdingbar ist. Aus diesem Grund wurden deterministische Verfahren entwickelt die ‚Pseudozufallszahlen‘ errechnen. Dabei werden Sequenzen von errechneten Zufallszahlen stets wiederholt, um so den Zufall nur zu simulieren, siehe z.B. (CT02). Jedoch werden die Sequenzen so konzipiert, dass statistisch kein Muster zu erkennen ist. Bei der Programmierung in Statistik-Software wie R, beziehungsweise RStudio (R C18), entspricht dies dem festgelegten ‚seed‘, einem Startwert, den man mit dem Befehl *seed*() und einer frei gewählten Zahl festlegt. Auf Basis dieses seeds wird dann die zu wiederholende Sequenz gebildet. Um nun zunächst gleichverteilte Zufallszahlen auf dem Intervall [0,1] zu erzeugen, erfand von Neumann 1946 die Mittelquadratmethode. Dabei wird von einer gewählten Anfangszahl, dem seed, zunächst das Quadrat gebildet. Nun werden aus der resultierenden Zahlenfolge die mittleren Ziffern benutzt, um das erste Output, die Zufallszahl, zu ergeben. Dieses Output wird nun als neuer seed verwendet aus dem wiederum ein Quadrat gebildet wird. Dies wird so weit wie gewünscht iteriert, siehe z.B. (HK19). Wie in Abbildung 3.1 zu sehen am Beispiel des seeds 798269. Diese Überlegung von Neumanns wird heutzutage jedoch nicht mehr benutzt, da es nur kurze Sequenzen erzeugen kann, siehe z.B. (HK19). Dennoch stellte diese Methode den ersten Pseudo Random Number Generator (PRNG), den Pseudozufallszahlengenerator dar. Somit konnten nun deterministische Zufallszahlen mit den Eigenschaften einer Folge unabhängig identisch verteilter Zufallszahlen erzeugt werden, siehe z.B. (Obs13). Zu den heutzutage bekanntesten und meistverwendeten Zufallszahlengeneratoren zählen die Kongruenzgeneratoren. Diese hier verwendeten linearen und multiplikativen Kongruenzgeneratoren wurden 1949 von Derrick Henry Lehmer, ein ebenfalls amerikanischer Mathematiker, entwickelt. Die mathematische Berechnung dahinter besitzt die Parameter m

für den Modulo, also den Divisionsrest, einen Faktor a, den Steigungsparameter b und den Startwert X_0:

$$X_{n+1} = (aX_n + b) \bmod m, \tag{3.1}$$

falls $n \geq 0$, $0 < m$, $0 \leq a < m$, $0 \leq b < m$ und $0 \leq X_0 \leq m$. Wobei m, n natürliche Zahlen sind. Es ergeben sich der lineare Sonderfall bei $n = 1$ und bei $b = 0$ der multiplikative Kongruenzgenerator. Das Ergebnis der X_{n+1} ergeben die Zufallszahlen welche aus wiederum dem vorhergegangenen X_n determiniert wurde. Nach m^n Durchführungen ist die Sequenz vorüber und sie muss wiederholt werden, siehe z.B. (Knu97). So hängt also die erzeugte Zufallszahlenfolge vom gewählten seed ab. Diese Zahlenfolge wiederholt sich nach einer gewissen Anzahl an Aufrufen. Dies wird als die Periode des Zufallszahlengenerators bezeichnet.

Nach Beobachtung der Monte-Carlo Algorithmen stellten die Wissenschaftler fest, dass einige Simulationen notwendigerweise eine Illustration von Markov-Ketten sein mussten (Wat11). Markov-Ketten, beziehungsweise Markov-Chains sind Abfolgen von Ereignissen, welche, je nach Ordnung der Markov-Kette, in einem wahrscheinlichkeitsbedingten Zusammenhang stehen. Genauer gesagt, ein zukünftiger wird durch den aktuellen Zustand beeinflusst, jedoch nur durch diesen, denn Markov-Ketten sind gedächtnislos. Das heißt, die Ereignisse aus der Vergangenheit beeinflussen zukünftige Zustände nicht. Eine Kette n-ter Ordnung beschreibt eine Kette mit n-vergangenen Zuständen, siehe z.B. (Hem19). Mathematisch ausgedrückt, heißt ein zeitlich diskreter Prozess

$$Y = \{Y_t, t \in \ N_0\} \tag{3.2}$$

mit abzählbarem Zustandsraum S eine Markov-Kette, falls

$$P\left(Y_t = k | Y_0 = j_0, Y_1 = j_1, ..., Y_{t-1} = j_{t-1}\right) = P\left(Y_t = k | Y_{t-1} = j_{t-1}\right) \tag{3.3}$$

für alle $t \geq 0$ und $k, j_0, ..., j_{t-1} \in S$ gilt. Die Übergangswahrscheinlichkeit zeigt sich wie folgt:

$$P\left(Y_{t+1} = k | Y_t = j\right) \tag{3.4}$$

(Wag11). Das einfachste Beispiel dafür sind Wetterprognosen. Man stimmt intuitiv zu, dass wenn heute beispielsweiße die Sonne scheint, die Wahrscheinlichkeit, dass sie morgen wieder scheint höher ist, als dass es Regen- oder Schneefall gibt. Schnell vermutet man eine Verwandtschaft mit der Bayesschen Statistik, welche der Mathematiker schon im 18. Jahrhundert entwickelte. Diese hilft bei der Berechnung bedingter Wahrscheinlichkeiten aufgrund von a-priori Vermutungen, was durchaus eine Ähnlichkeit zu den MCMC Methoden aufweist. Nicholas Metropolis, ein amerikanischer Physiker der ebenfalls am Manhattan Projekt beteiligt war, kombinierte und optimierte nun die Monte-Carlo und Markov-Chain Schritte und veröffentlichte in seiner Arbeit „Equation of State Calculations by Fast Computing Machines“ (MRR$^+$53), was heute als der erste MCMC-Algorithmus angesehen werden kann, den Metropolis-Algorithmus (RC11). Der besondere Nutzen der Kombination beider Anwendungen ist es zu ermöglichen, eine a-posteriori Verteilung, wie sie durch den Satz von Bayes berechnet wird, zu approximieren. Um also einen Wert des posteriori Zustandes zu schätzen

werden anhand eines zufälligen Wertes, somit die a-priori Annahme, mit Hilfe der Monte-Carlo Methode Pseudozufallswerte erzeugt, also Werte mit einer zufälligen Ähnlichkeit zum anfänglichen Wert. Beschreibt einer dieser zufälligen Werte den a-priori Wert hinreichend gut, wird er mit einer bestimmten Wahrscheinlichkeit zur Kette der Werte hinzugefügt, was die Aufgabe des Markov-Chain Anteils ist. Auf diese Weise entsteht eine Schätzung über die wahre a-posteriore Verteilung, siehe z.B. (Hem19). Diese Entdeckung führte zu einem Wendepunkt in der Statistik. Zusammen mit der Entwicklung eines der ersten Computer ENIAC, siehe auch Kapitel 2, eröffneten sie neue Möglichkeiten, nicht nur in der Statistik. Metropolis fasste nach (Wat11) die Auswirkungen des ENIAC folgendermaßen zusammen: "Sampling techniques were well known to statisticians but had not enjoyed much application because of the tedium of implementing the various concepts. [... Stanislaw] Ulam realized that [electronic computers] had changed all that [, and that ...] sampling could be trivialized and statistical techniques should be revived. [...] Plans were made to test the method on various nuclear material configurations to determine criticality, spatial and velocity distributions of neutrons, and so forth using the ENIAC." Auch im anfänglichen Zusammenhang mit dem Manhattan Projekt ergaben sich innovative Anwendungsmöglichkeiten für die experimentelle Physik. Wo bisher zufällige Verteilung mühevoll durch beispielsweise die Messung von radioaktivem Zerfall mithilfe eines Geigerzählers simuliert wurde, erkannten die Forscher die Vorteile und die Einfachheit der MCMC-Algorithmen. Noch in den 1950er Jahren hatte jeder Computer einen PRNG verbaut oder hatte Zugriff auf dementsprechende Tabellen. Jedoch konnte dieser Mehrwert für die Statistik vorerst noch nicht von jedem genutzt werden, da Computer schließlich noch nicht für jeden zugänglich, geschweige denn bezahlbar waren (Wat11). Außerdem wurde Vorwissen, wie das Programmieren mit FORTRAN benötigt, welches zu dieser Zeit noch nicht benutzerfreundlich und schwer zu lernen war, siehe auch Abschnitt 2.3. Dies waren Gründe wodurch MCMC Methoden erst Jahrzehnte später ihren eigentlichen Durchbruch schafften und den Weg in den alltäglichen Gebrauch der Statistik fanden.

3.2 Die zweite Revolution: Sampling und Resampling

Gibbs Sampling

> „The prodigious advances made by Bayesian analysis in methodological and applied directions during the previous decade have been made possible only by advances of the same scale in computing abilities with, at the forefront, Markov chain Monte Carlo methods [...]. Many things happened in Bayesian analysis because of MCMC and, conversely many features of MCMC are only there because of Bayesian analysis! We think the current state of Bayesian analysis would not have been reached without MCMC techniques and also that the upward surge in the level of complexity of the models analyzed by Bayesian methods contributed to the very fast improvement in MCMC methods."

(RDA04)

Dieses Zitat der Mathematiker Christophe Andrieu, Arnaud Doucet und Christian Robert beschreibt die weitere Entwicklung in der Statistik nach der Entdeckung der MCMC-Verfahren sehr gut (RDA04). Trotz schleppender Eingliederung dieser neuen Verfahren und der Bayesschen Überlegungen in die Statistik entdeckten einige Mathematiker und Statistiker das Potential und machten sich daran diese neuen Methoden zu nutzen und weiter zu entwickeln. Dies fand jedoch erst in den 90er Jahren seinen Höhepunkt, als Alan E. Gelfand, ein amerikanischer, und Adrian F. M. Smith, ein britischer Statistiker, ihren Artikel „Sampling-Based Approaches to Calculating Marginal Densities“ in dem „Journal of the American Statistical Association“ veröffentlichten (GS90). Ihre Veranschaulichung des Gibbs-Samplers als Spezialfall des Metropolis-Algorithmus machte komplizierte Bayes Statistiken möglich. Die Brüder Geman führten schon 1984 das Gibbs-Sampling in ihrem Artikel „Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images“ ein (GG84). Jedoch erst in Verbindung mit der beständigen Etablierung des Computers im Alltag und der stetig ansteigenden Rechenleistung entwickelte sich laut Christian Robert und George Casella, eine “second-generation MCMC revolution”, also eine zweite MCMC-Revolution. Begleitet von der ersten Software auf dem Themengebiet, der BUGS („Bayesian inferece Using Gibbs Sampling“) aus dem Jahr 1991, wurden viele Wissenschaftler von der Einfachheit der Methode überzeugt. Neues Interesse an Bayesschen Methoden, statistischem Rechnen, Algorithmen und stochastischen Prozessen durch den Einsatz von Rechenalgorithmen wie dem Gibbs-Sampler und dem Metropolis-Algorithmus wurde geweckt und kam so in der massentauglichen Statistik an, siehe z.B. (RC11).

Das Gibbs-Sampling, aufgrund von Einflüssen aus der statistischen Physik nach dem Physiker Josiah Willard Gibbs benannt, ist eine weiterentwickelte Methode der MCMC Verfahren. Der Nutzen liegt in der Möglichkeit Zufallszahlen durch die bekannten Eigenschaften der Markov-Ketten aus unbekannten Verteilungen zu ziehen. Genauer, aus einer unbekannten gemeinsamen Verteilung $f(x, y_1, ..., y_p)$ mit bekannter bedingter Verteilung zweier oder mehrerer Zufallsvariablen die Zufallszahlen $X_1, ..., X_m \sim f(x)$ zu approximieren, denn diese konvergieren gegen die theoretischen Werte. So wird beim Gibbs-Sampling abwechselnd aus den bedingten Dichten

$$X_j^{'} \sim f(x|Y_j^{'} = y_j^{'}) \tag{3.5}$$

$$Y_{j+1}^{'} \sim f(y|X_j^{'} = y_j^{'}), \tag{3.6}$$

mit Startwert $Y'_0 = y'_0$ gezogen. So entsteht die Gibbs Folge

$$X_0^{'}, X_1^{'}, X_2^{'}, ..., X_n^{'} \tag{3.7}$$

mit $n \to \infty$. Diese ergibt sich aber erst aus der Folge $X_0^{'} \to Y_1^{'} \to X_1^{'}$, womit $X_0^{'} \to X_1^{'}$ eine Markov-Kette mit Übergangswahrscheinlichkeit

$$P\left[X_1^{'} = x_1|X_0^{'} = x_0\right] = \sum_y P\left[X_1^{'} = x_1|Y_1^{'} = y\right] \times P\left[Y_1^{'} = y|X_0^{'} = x_0\right] \tag{3.8}$$

darstellt. $X_n^{'} = x_n^{'}$ sind somit die Zufallszahlen aus $f(x)$. Mit $k \to \infty$ konvergiert die Verteilung von $X_k^{'}$ nun gegen die von f_x, siehe z.B. (Obs13).

Von da an begann ein großer Aufruhr und die Verbreitung der neuen Möglichkeiten für Simulation und Simulationsstudien nahm seinen Lauf. Plötzlich konnte man praktisch unlösbare Probleme und Modelle empirisch simulieren. Im Februar 1991 wurde unter der Leitung von Alan Gelfand, Adrian Smith und Prem Goel, eine Konferenz mit Workshop zur Bayesschen Berechnung mittels stochastischer Simulation abgehalten, welcher viel Forschung und Begeisterung nach sich zog. Aus den Vorträgen entstandene Paper und Veröffentlichungen haben bis heute eine große Bedeutung für die Statistik. Einige der Themen waren: Theoretischer Aspekt des iterativen Sampling von Adrian Smith, Posteriore Simulation und Markov-Sampling von Gelfand, Adaptives Sampling von Carl Morris, Professor der Statistik an der Harvard Universität, Generalisierte lineare und nichtlineare Modelle durch Rob Kass sowie Maximimum-Likelihood (ML) und gewichtetes Bootstrapping von George Casalla (RC11). Weitere Konferenzen fanden statt und auch die Royal Statistical Society, eine Gesellschaft für Statistiker in Großbritannien, beteiligte sich.

Bootstrap und Jackknife

John W. Tukey, ein amerikanischer Statistiker, stellte schon 1962 in seinem Werk: „The Future of Data Analysis“ fest:

> „[. . .] there are situation where the computer makes feasible what would have been wholly unfeasible. [. . . S]peed and economy of delivery of answer make the computer essential for large data sets and very valuable for small sets.”

(Tuk62). Auch seine Entdeckung, Tukeys Jacknife, setzte sich erst durch die voranschreitende Computerentwicklung richtig durch. So sagte er weiter inhaltsgemäß: „Es ist übrigens sowohl überraschend als auch unglücklich, dass die mit statistischer Theorie und statistischer Mathematik befassten Personen so wenig Kontakt zu empirischen Stichproben hatten. Man stellt fest, dass nicht mehr als ein paar hundert Stichproben ausreichen, um die meisten Fragen mit ausreichender Genauigkeit zu beantworten und bei schnellen elektronischen Maschinen stellen solche keinen großen Zeit- oder Geldaufwand dar.“ (Tuk62). Tukey hat viele Bereiche in der Statistik sowie der Informatik geprägt. So war er der Erste, der das Wort ‚Software‘ als Gegenstück zur ‚Hardware‘ verwendete. Auch das Wort ‚Bit‘, einer Zusammensetzung der Wörter ‚binary‘ und ‚digit‘, haben wir ihm zu verdanken. Des Weiteren begründete er die Explorative Statistik und vor allem den Boxplot und das Stamm-Blatt-Diagramm. Dazu sagte er im Journal American Statistician: „As yet I know of no person or group that is taking nearly adequate advantage of the graphical potentialities of the computer. [. . .] In exploration they are going to be the data analyst's greatest single source.” (Tuk65).

So wurden nun auch die Vorteile der immer leistungsfähigeren Computer genutzt und Resampling Techniken wie Bootstrapping-Verfahren, die Jackknife-Methode, Permutationstests oder Kreuzvalidierung fanden Einzug in die neue statistische Praxis. Auch wenn Resampling seit den 1980er Jahren besteht, war der Aufwand ohne Rechnerunterstützung sehr groß, wenn nicht gänzlich unmöglich. Denn unter diesem Begriff versteht man eine wiederholte Stichprobenziehung mit Zurücklegen. Im Fall von Bootstrapping handelt es sich um ein nonparametrisches Verfahren für den Fall einer unbekannten Verteilung oder wenn die Normalverteilungsannahme nicht sicher erfüllt ist, es wird also keine Verteilungsannahme getroffen.

So zieht man n Bootstrap-Stichproben $x_b = (x_1^{'}, ..., x_n^{'})$ mit $b = 1, ..., B$ aus einer ursprünglichen Stichprobe mit Zurücklegen. Also können Auslassungen oder Mehrfachziehungen entstehen. Dies imitiert das Ziehen von Zufallszahlen aus einer empirischen Verteilung. Daraufhin wird die interessierende Statistik $\hat{\theta}^{'}$ einer jeden Bootstrap-Stichprobe errechnet, etwa der Mittelwert oder Odds-Ratio, und die Verteilung durch diese Werte $\hat{\theta}_1^{'}, \hat{\theta}_2^{'}, \ldots, \hat{\theta}_B^{'}$ approximiert. Desweiteren kann man daraus Tests oder Konfidenzbereiche konstruieren (SN19). Ein Spezialfall des Bootstrapping ist die schon erwähnte Jackknife-Methode von John Tukey (YR86). Dabei werden wiederholt Stichproben aus einer Quelle gezogen, jedoch wird bei jeder Stichprobenziehung ein Wert aus der ursprünglichen Stichprobe weggelassen und daraus dann der Schätzer berechnet. Deswegen wird oft analog der Begriff ‚delete-1-Jackknife' verwendet. Allgemein, oder wenn nicht nur ein Wert sondern d Werte pro Stichprobenziehung ausgeklammert werden, nennt man den Algorithmus entsprechend ‚delete-d-Jackknife'. Diese Vorgehensweise dient zur Berechnung der Qualität von Schätzwerten, woraus letztendlich der zufällige Fehler und Verzerrungen errechnet werden können. Um nun den Schätzer $\hat{\theta}$ zu berechnen, werden n Werte $\hat{\theta}_{(i)}$ wie folgt iteriert: Für $\hat{\theta}_{(1)}$ wird die erste Zufallsvariable X_1 weggelassen. Die erste reduzierte Stichprobe lautet demnach $(X_2, X_3, \ldots, X_n)$ mit $n-1$ Werten. Zur Berechnung von $\hat{\theta}_{(2)}$ wird der zweite Wert $\hat{\theta}_{(2)}$ ausgeschlossen und die Stichprobe lautet $(X_1, X_3, \ldots, X_n)$. Nun berechnet man den Mittelwert der errechneten Stichproben:

$$\hat{\bar{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_{(i)}. \tag{3.9}$$

Der Bias ergibt sich dann durch

$$\hat{Bias} = (n-1)(\hat{\bar{\theta}} - \hat{\theta}). \tag{3.10}$$

Die Schätzung des Standardfehlers wird wie folgt berechnet:

$$\hat{se} = \left[\frac{n-1}{n} \sum_{i=1}^{n} (\hat{\theta}_{(i)} - \hat{\bar{\theta}})^2 \right]^{\frac{1}{2}}. \tag{3.11}$$

Dabei wird $\dfrac{n-1}{n}$ als Inflationsfaktor bezeichnet. Denn jede Jackknife Stichprobe umfasst $n-1$ Beobachtungen bei n verschiedenen Stichproben, siehe z.B. (Ste11).

Wie sich erkennen lässt, benötigt man, beispielsweiße bei der Berechnung der Standardabweichung, bei der Jackknife Methode weniger Rechenschritte als durch Bootstrapping. Jedoch ist der Bootstrap zur Berechnung von Konfidenzintervallen sehr nützlich, da es eine Verteilung der Schätzstatistik generiert, siehe z.B. (Kna07).

Kreuzvalidierung

Auch die Kreuzvalidierung ist eine Methode, die schon weit vor der Einführung des Computers in die Wissenschaft entwickelt wurde (Obs13), jedoch erst dadurch Einzug in die Praxis gehalten hat. Denn dieser Algorithmus kann ein sehr rechenintensiver Vorgang sein. Um die Anpassungsgüte eines Modells zu bestimmen gibt es mehrere verschiedene Techniken.

Die wohl bekannteste ist die k-fold- , oder zu deutsch die k-fache-Kreuzvalidierung. Dabei teilt man einen Datensatz in k gleich große Teildatensätze auf. Zunächst werden dann $k-1$ Datensätze, auch Trainingsdatensätze genannt, im Modell geschätzt. Der k-te Datensatz, oder auch Testdatensatz, wird zur Validierung des Modells verwendet. Dies wird iteriert bis jeder der $1:k$ Teile einmal als Validierungsdatensatz fungiert hat. Durch die k errechneten Testfehler wird ein durchschnittlicher kreuzvalidierter Fehler der Vorhersage des Modells erstellt. Es gibt noch weitere Techniken, wie die leave-one-out-Kreuzvalidierung, welche dem Vorgehen des Jackknife ähnelt, sowie die holdout-Kreuzvalidierung, bei der man Trainings- und Testdatensatz in zwei Teile mit gewähltem Verhältnis teilt.

Wenn man diese Entwicklung von den frühen 1940er Jahren während des zweiten Weltkriegs und einer zunächst wohl unbedeutend wirkenden Überlegung beim Kartenspiel über zu Sampling und Resampling betrachtet, gepaart mit den rasanten Entwicklungen des Computers und der Rechenleistung, ist es nicht überraschend, dass in der weiteren Zukunft der computationalen Statistik die Möglichkeiten exponentiell anstiegen. Immer mehr Wissenschaftler nahmen die Techniken an und brachten ihr Wissen ein. So sagte John Ashworth Nelder, ein britischer Mathematiker und Statistiker schon 1984 und prophezeite:

> "One thing seems certain: any statistician who seeks to influence the development of our subject in the next 150 years must become involved with computers."

(Wat11).

3.3 The R Project

> "It is statistical software that has revolutionized the way we approach data analysis, replacing the calculators - mechanical, electrical, electronic - used by earlier generations."

Brian David Ripley, ein britischer Mathematiker und Professor an der University of Oxford, machte dieses Zitat (Rip05) und beschrieb treffend, in welche Richtung die weitere Entwicklung der Statistik ging. Denn neben den Neuerungen in der Statistischen Methodik wurde parallel auch daran gearbeitet, wie man diese neuen Methoden noch weiter vereinfachen und auch für jeden ausführbar bereitstellen konnte. Dies gewann insbesondere an Bedeutung zu einer Zeit in der der Heimcomputer für die breite Bevölkerung bezahlbar wurde.

Von den Anfängen der Programmiersprachen bis zum R Project

Alles begann mit der schon erwähnten, ersten höheren Programmiersprache FORTRAN. Diese wurde bereits 1954 von dem Informatiker John Backus entwickelt. Er ist bekannt durch die Backus-Naur-Form, welche die formalen Regeln beschreiben, unter denen eine höhere Programmiersprache funktioniert. FORTRAN, später auch Fortran geschrieben, ist heute eine objektorientierte und prozedurale Sprache, welche es Programmierern zunächst vereinfachen sollte numerische Probleme und Formeln zu implementieren. Schnell zeigten sich die Vorteile und Möglichkeiten für das Sampling, wie in Abschnitt 3.2 beschrieben (Sla17).

FORTRAN wird bis heute ständig verbessert und weiter entwickelt. Jeder neuen Version wird das Erscheinungsjahr angefügt. Die neueste heißt Fortran 2018.

Zu Beginn der 1970er Jahre wurde dann die Programmiersprache „C“ von Dennis Ritchie aus ihrem Vorgänger, der Sprache „B“, entwickelt. Ritchie arbeitete in den Bell Laboratories, einer Forschungseinrichtung der früheren AT&T Telefongesellschaft, welche durch ihre bahnbrechende Forschungsarbeit einen großen Gewinn für die Mathematik, Physik und Informatik darstellte. C wurde als Systemimplementierungssprache für das Unix Betriebssystems entwickelt. Dies war ein freies und quelloffenes Betriebssystem, das heißt es war anfangs möglich und sogar gewollt, dass andere Programmierer und Nutzer beim Quellcode mitwirken. Dies wurde jedoch in den 1980er Jahren von AT&T unterbunden, um Lizenzgebühren verlangen zu können. Das sorgte in der Programmierer-Gesellschaft für viel Unmut (Rit96). So rief einer von ihnen, Richard Stallman, 1983 das GNU's Not Unix (GNU)-Projekt ins Leben, mit dem Ziel ein freies Betriebssystem zu schaffen, welches zudem Unix-kompatibel ist. Sein Wunsch war es Software zu schaffen, welche jeder bei Bedarf verändern, kopieren sowie untersuchen und studieren konnte. So wurden im Sinne der GNU General Public License (GPL), der GNU Veröffentlichungsgenehmigung, eine stetig wachsende Sammlung freier Software, bestehend aus Anwendungen, Bibliotheken, Extras für Entwickler und Spiele erstellt (Fre19). Im Zuge dessen entstand 1991 auch das freie Betriebssystem Linux durch den Software-Entwickler Linus Torvalds. Die nächste für die Statistik bedeutsame Sprache kam 1976 unter dem Namen „S“ auf. Die Informatiker und Statistiker Richard A. Becker, John M. Chambers und Allan R. Wilks entwickelten diese interaktive und objektorientierte Programmiersprache an den Bell Laboratories. S vereinfachte die bisher sehr komplizierte Datenanalyse durch FORTRAN. Es erleichterte vor allem die Darstellung und das Generieren von Grafiken und die explorative Datenanalyse. Denn interaktiv bedeutet in diesem Fall, dass der Nutzer nicht erst ein Programm kompilieren muss, sondern direkte Ergebnisse erhält, siehe z.B. (Obs13). Mit den verschiedenen Versionen von S, S1 bis S4, wurden immer neue Standards in der statistischen Programmierung festgelegt. Während S1 lediglich numerische Probleme, Zufallszahlengenerator und Modelle beherrschte, wurden in S2 gegen 1980 bereits Klassen und Objekte eingeführt. Ab 1988 herrschte dann das S3-Klassensystem vor, mit dem man nun Funktionen formulieren konnte. Noch bevor 1998 S4 mit einer Erweiterung des S3 Objekt- und Klassenmodell eingeführt wurde, wurde eine kommerziellere Version herausgegeben, S-PLUS.

Und dieser Schritt führte schließlich dazu, dass die zwei Statistiker Ross Ihaka und Robert Gentleman 1992 an der Universität Auckland mit ihrer Arbeit an einer freien Version der Sprache S begannen, die Geburtsstunde von R. Der Name ist eine Hommage an S, aber sie benutzten ihrer beider Anfangsbuchstaben. Auch gefiel ihnen die Tatsache, dass man einen einfachen Buchstaben nicht urheberrechtlich schützen könne, siehe z.B. (Put10). Anfangs noch geheim gehalten, wollten sie die für gut befundene Syntax von S weiterentwickeln und um einige Funktionen, vor allem Benutzerfreundlichkeit, erweitern, siehe z.B. (Neu18). Somit ist R eine freie Implementierung von S. Sie benutzten C und FORTRAN für die Programmierung. 1993 veröffentlichten sie dann ihre unfertige Version von R in einem Forum für statistische Software. Dies fand so viel Anklang, dass in kürzester Zeit mit der Hilfe anderer Statistiker und Informatiker eine fertige Programmiersprache daraus wurde. Da es

bis zu diesem Zeitpunkt keine freie und effiziente statistische Programmiersprache gab, war die Begeisterung groß. 1995 wurde R dann auch Teil des GNU-Projekts und unter der GPL lizensiert (Neu18). Ross Ihaka ist letztendlich froh über diesen Schritt: "We could have made it a commercial thing and five people would have used it. But given that we made it freely available, people added so much value to it. Unless we had actually made it free software, nobody would have contributed because they would be asking 'well, where is my piece of the action'" (Put10).

Im Jahr 1997 gründete die bisher kleine Gruppe um das R Project das R Development Core Team. Heute lediglich R Core Team genannt, kümmern sie sich um den Quellcode und die Weiterentwicklung. Einer der großen Vorteile, und sicherlich ein Grund warum R heute so beliebt ist, ist die Möglichkeit der Pakete. Im Sinne der GNU Philosophie haben Anwender selbst die Möglichkeit eigene Pakete zu erstellen und allen anderen zur Verfügung zu stellen. Dies unterliegt natürlich einigen Qualitätskriterien. Jedoch kann sich R seit vielen Jahren in der Statistiker Gemeinde behaupten, da es die Pakete stets auf dem neuesten Stand der Statistik bringt. Dazu wurde 1997 das Comprehensive R Archive Network (CRAN) als Plattform für alle Pakete gegründet. Es ist somit ein Netzwerk von Servern die aktuelle Code- und Dokumentationsversionen für R speichern. Chris Triggs, Statistik Professor an der Geburtsstädte des R, der University of Auckland, beschreibt diese Möglichkeit mit Stolz: "Let's suppose I invent a new method of doing a particular calculation or analysing data from a particular problem. Nobody else is going to use it unless they are given the computer program that does the analysis. So as well as doing the theoretical development and validation, somebody making an innovation will write an R program or an R function to do the analysis." (Put10).

In den Jahren darauf wurden neben der Version für Unix-Betriebssysteme, auch die für Microsoft Windows und macOS vorbereitet. Und schließlich, am 29. Februar 2000 wurde die erste, vom R Development Core Team als stabil betrachtete Version, R 1.0.0 mit 12 Paketen veröffentlicht (The19).

Das R Project heute

Heute, fast 27 Jahre später ist R eine der meist verbreitetsten Programmiersprachen und für Datenwissenschaftler, Mathematiker oder Statistiker kaum wegzudenken. Vor allem dadurch, dass R eine freie Software ist, ist der Zugang für Universitäten und deren Studenten denkbar einfach und hat unter anderem deswegen auch die Lehre bedeutend mitgeprägt. Die eigene Skriptsprache erfordert keine Vorkenntnisse in der Programmierung und die benutzerfreundliche Entwicklungsumgebung von RStudio ermöglicht ein einfaches Bedienen der Funktionen, siehe z.B. (Neu18). So wuchs der Funktionsumfang über die Jahre mit der Versionsnummer. Diese ist durch drei Zahlen gegliedert, eine grundlegende Änderung erhöht die erste Zahl, eine Änderung geringeren Ausmaßes erhöht die mittlere Zahl und kleine Bug Fixes, zu deutsch Fehlerbehebungen, ändern die letzte Zahl. So wurde beispielsweise Ende 2003 mit Version 1.8.0 das Paket *grid* hinzugefügt um Grafiken besser bearbeiten zu können, 2004 wurde bei Version 2.0.0 das sogenannte Lazy Loading eingeführt, was zu schnelleren Ladeergebnissen führt, und unter anderem das Paket *datasets* addiert, welches Beispielda-

tensätze beinhaltet. Seit 2011 und 2.14.0 erhalten die Versionen auch Beinamen passend zur Jahreszeit oder aktuellen Ereignissen, so heißt 2.14.0 „Great Pumpkin“ oder 3.4.2 „Short Summer“. Die aktuelle Version vom 05.07.2019 heißt 3.6.1 „Action of the Toes“ (The19).

Im Jahr 2000 wurde die gemeinnützige Stiftung, die R Foundation, gegründet. Diese hilft bei der Weiterentwicklung von R, siehe z.B. (Put10). Finanziers sind dabei unter anderen AT&T Research sowie Google. Die selbst gesetzten Ziele der Foundation sind es stets neue Methoden zu erforschen, die Lehre und Ausbildung in der statistischen Datenverarbeitung und Workshops sowie Konferenzen in diesem Fachbereich zu unterstützen. Außerdem ist sie Teil der GNU Free Software Foundation (FSF), einer weltweiten Mission zur Förderung der Freiheit von Softwareanwendung. Seit 2004 hält die R Foundation auch regelmäßige Konferenzen ab. Als wichtigste ist dabei die useR! – International R User Conference zu nennen. Dabei soll vor allem der R Nutzer angesprochen werden. Die useR! fand erstmals 2004 an der Technischen Universität Wien statt. Die Hauptpunkte waren einen Überblick über die neuen Funktionen und Projekte zu geben, das damals neue *grid* Paket, dem neuen S4 Klassensystem oder dem Vorgehen bei großen Datensätzen. Auch sollte es eine Plattform zum Austausch und für Diskussionen sein. Die nächste useR! wurde erst 2006, ebenfalls in Wien, abgehalten und seitdem wird sie jährlich realisiert. Im Jahr 2008 fand sie das erste und bisher einzige Mal in Deutschland, an der Technischen Universität Dortmund, statt. Zusätzlich zu weiteren kleineren Konferenzen wie den „R Day“, „LatinR“ oder „ConectaR“ veranstaltet die R Foundation die DSC - Directions in Statistical Computing. Hierbei geht es nicht rein um R, viel mehr um statistischer Software und die Forschung im Bereich der statistischen Datenverarbeitung allgemein sowie der Entwicklung statistischer Software. Die DSC fand erstmals 1999 ebenfalls in Wien statt, jedoch erst ab 2014 jährlich in diversen Städten. Siehe dazu (DSC14) und (The19).

Um die R News, den Newsletter für die R Gemeinde, aufzuwerten, wurden diese 2008 durch das R Journal ersetzt. Dies erscheint zweimal pro Jahr und beinhaltet kurze Artikel zu Themen rund um das Benutzen und Weiterentwickeln von R. Dabei wird darauf Wert gelegt, dass das Niveau nicht zu hoch liegt und ein breites Publikum erreicht wird. Es wird auch über die neuesten Versionen und Pakete sowie bevorstehender und vergangener Konferenzen berichtet (The19).

Als weiterer Pfeiler des R Projects ist das 2015 gegründete R Consortium zu nennen. Dieses umfasst Unternehmen welche R bei sich als Standardsprache eingeführt haben. Ziel ist es die geschäftliche Infrastruktur von R zu warten, verteilen und zu nutzen. Zusammen mit der R Foundation und RStudio beteiligen sich auch Biotech-, Finanz- und Forschungsunternehmen wie Microsoft, Google, Hewlett-Peckard, IBM, Tibco oder Oracle. Dabei werden einige Projekte zur Implementierung von R in Unternehmensprozesse unterstützt und gefördert, um die Verbreitung noch weiter auszubauen und für immer mehr Unternehmen attraktiv zu machen (R c19).

Auch wenn R eine der meist genutzten Programmiersprachen für statistische Auswertungen, Grafiken und Simulationen ist und das Output Publikationsqualität besitzt, sollte man auch auf andere statistische Software eingehen und deren Vorteile kennen. Obwohl die kostenfreie Anschaffung von R ein Vorteil ist, setzten viele Unternehmen oder Wissenschaftsbereiche auf andere Software. So ist Phyton in den letzten Jahren zu einer Alternative

besonders in Bereichen des Deep- und Machine-Learnings geworden. Es handelt sich dabei auch um eine eigene Programmiersprache welche in Anwendungssoftwarepaketen eingebettet, im englischen embedded, werden kann. Die Aktualität wird durch neue Pakete gewährleistet. Jedoch sind noch nicht alle statistischen Methoden vorhanden oder stabil. Eine weitere Software die vor allem in der biomedizinischen Statistik, klinischer Forschung und im Bankensektor standartmäßig angewendet wird ist SAS. Die Einarbeitung ist herausfordernd, jedoch im Pharma-Bereich unabdingbar, da SAS die Regeln der amerikanischen Arzneimittelbehörde und somit die Bedingungen für Studien erfüllt. Im Gegensatz dazu braucht man für die Software SPSS (Statistical Package for the Social Sciences) keinerlei Programmierkenntnisse aufgrund ihrer weit entwickelten GUI, also der Benutzeroberfläche mit grafischen Schaltflächen und Steuerelementen. Auch muss man nicht immer über die vollständigen Methodenkenntnisse verfügen. Deswegen ist SPSS eines der beliebtesten Programme unter Anderem in den Sozialwissenschaften und der Psychologie. Zuletzt ist noch STATA zu nennen, welches besonders in der Ökonometrie eingesetzt wird. Hier wird es benutzt für Panel Daten oder Strukturgleichungsmodelle (SEM) und ist somit besonders nützlich in den Wirtschaftswissenschaften. STATA besitzt auch entsprechende Steuerelemente jedoch ist eine Eingabe via Programmcode möglich. Dabei ist nahezu jede statistische Methode möglich, siehe z.B. (Gho18) und (Gr9). Zur Freude des Benutzers bieten mittlerweile viele Programme und Software die Möglichkeit auf R Funktionen zuzugreifen und den R Code zu implementieren was die Benutzerfreundlichkeit erhöht und einfacher zwischen verschiedenen Bereichen der statistischen Auswertung je nach Vorliebe und Bedürfnis variieren lässt, siehe z.B. (Neu18).

Bisher ist sicher zu sagen, dass der Schritt, aus der Programmiersprache S eine freie und benutzerfreundliche Sprache zu entwickeln die Statistik allgemein sowie den computationalen Aspekt in ihrer Entwicklung sehr geholfen hat. R machte möglich, dass alle Studenten Zugang zu R und daraufhin RStudio haben und die Universitäten praxisbezogenen Unterricht im Sinne der computergestützten Statistik anbieten können. Wie man in den vorherigen Kapiteln erkennen konnte ist dies der Schlüssel in jeden möglichen Bereich der Statistik vorzudringen den man „per Hand“ nicht erreichen könnte. Würde es diese kostenfreie Möglichkeit nicht geben, müssten Universitäten und Studenten oft sehr hochpreisige Lizenzen anderer Software erwerben. Ein weiterer Aspekt, der daraus resultiert ist, dass die Unternehmen in dem Wissen, dass jedes Jahr Studenten mit R-Kenntnissen die Universitäten verlassen, sich dementsprechend anpassen und somit viele schon R als ihre Standartprogrammiersprache realisiert haben. Wie man am R Consortium sieht, kann somit eine große Infrastruktur erreicht und immer weiter ausgebaut werden. So entsteht eine Lingua franca, eine Verkehrssprache, welche allen beteiligten Personen und Unternehmen die Kommunikation erleichtert und auch die Möglichkeit besitzt zwischen verschiedenen Fachbereichen zu vermitteln. Nicht zuletzt durch die einfache Erweiterung in die verschiedensten Gebiete der Datenwissenschaft durch immer neue Pakete, passt sich R leicht an jedes Bedürfnis an.

3.4 Die Künstliche Intelligenz

> “At the root of intelligence are symbols, with their denotative power and their susceptibility to manipulation. And symbols can be manufactured of almost any-

> thing that can be arranged and patterned and combined. Intelligence is mind implemented by any patternable kind of matter.“

Die Vorstellung, dass Intelligenz Verstand ist, der durch jede modellierbare Art von Materie implementiert wird, wie Allen Newell und Herbert Alexander Simon nach (Chr00) in diesem Zitat ansprechen, zeigt den Grundgedanken hinter Machine Learning als Teilkonzept der Künstlichen Intelligenz. Die beiden US-Amerikaner aus den Bereichen der Elektrotechnik, Kognitionspsychologie und Sozialwissenschaft gelten als zwei der Väter der künstlichen Intelligenz. Doch zunächst ist eine genaue begriffliche Einteilung notwendig. Machine Learning ist ein Teilbereich der Künstliche Intelligenz (KI), oder des englischen Begriffes Artificial Intelligence (AI). Die KI bedeutet eine Problemlösung mithilfe von Computersystemen im Allgemeinen. Machine Learning meint dabei, dass diese Problemlösung neuer oder unbekannter Probleme selbstständig durch eine Maschine gelöst wird und zwar weil diese Wissen aus Erfahrung generiert hat, siehe z.B. (Man18). Das heißt die Mustererkennung aus Daten und die Verwendung dieser Muster um zukünftige Daten vorherzusagen oder Entscheidungen unter Unsicherheit durchzuführen. Somit kann man einem künstlichen System, wie einem Menschen auch, Lernen beibringen ebenso wie die daraus resultierende Anwendung des Gelernten. Dies ist eine noch junge Disziplin der Datenanalyse, jedoch entstanden aus den durch die computationale Wende ermöglichten algorithmischen Methoden, siehe z.B. (Obs13). Dabei hat die KI und damit auch das Machine Learning seine Anfänge natürlich auch in der Statistik. Beides verfolgt die Wissenschaft des Lernens aus Daten. Jedoch hat sich das maschinelle Lernen früh abgespalten und andere Intentionen mit eigener Kultur und Geschichte entwickelt (Was13). Leo Breiman, ein vielfach ausgezeichneter US-amerikanischer Statistiker und Professor, welcher die Bereiche der Klassifikationsverfahren mithilfe von Random Forrests und Entscheidungsbäumen prägte, leistete viel Forschungsarbeit an der Schnittstelle zwischen der Statistik und der Informatik. Er bezeichnete beispielsweise den Unterschied der Statistik zum maschinellen Lernen folgendermaßen: Zunächst einmal müsse man sich, seiner Überlegungen nach, Daten als von einer Blackbox erzeugt vorstellen. Wie in Abbildung 3.5 zu sehen, fungiert innerhalb der Blackbox die Natur als Verbindung zwischen der unabhängigen Variablen, dem Input, x und der Response Variable, der Output, y.

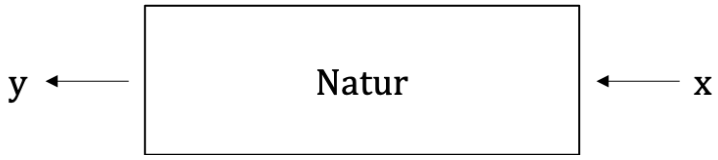

Abbildung 3.2: Die Verbindung der Kovariablen mit dem Response; Eigene Darstellung nach (Bre01)

Somit hat die Datenanalyse zwei Ziele. Zum einen das Ziel der Vorhersage, also die Antworten auf zukünftige Inputvariablen vorherzusagen. Zum anderen die Information, also wie die Natur die Zielgröße den Einflussgrößen zuordnet. Und um diese Ziele zu erreichen gibt es laut Breiman zwei verschiedene Ansätze.

Das ist zuerst die „Data Modeling Culture“ also die Datenmodellierung. Diese ist in Abbildung 3.3 zu sehen und stellt die Annahme der Statistik eines stochastischen Datenmodells im

Inneren der Blackbox dar. Diese lässt sich auch unter der Funktion: Response Variable $y = f(\text{Kovariablen } x, \text{zufälliger Fehler } \epsilon, \text{Parameter})$ darstellen. Die Parameter werden dabei aus den Daten geschätzt, der Response aus unabhängigen Ziehungen aus dem Modell f. Dies

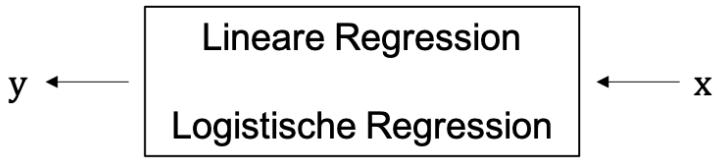

Abbildung 3.3: The Data Modeling Culture; Eigene Darstellung nach (Bre01)

stellt auch die gesuchte Information und Vorhersage dar. Über Hypothesentests und Signifikanz der Kovariablen kann im Sinne der jeweiligen Modellannahmen Information gezogen werden und mit Blick auf die Responsevariable Vorhersagen getroffen werden.

Der zweite Ansatz beschreibt die „Algorithmic Modeling Culture“ wie in Abbildung 3.4 zu sehen. Dies zeigt wie das maschinelle Lernen als Teilbereich der KI die Datenmodellie-

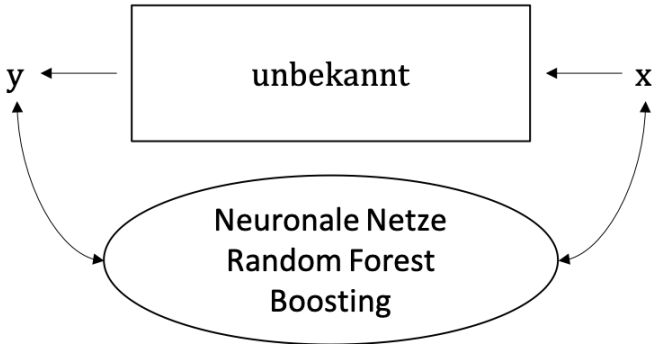

Abbildung 3.4: The Algorithmic Modeling Culture; Eigene Darstellung nach (Bre01)

rung betrachtet. Hier ist das Innere der Blackbox unbekannt und zudem komplex. Es wird versucht eine Funktion $f(x)$ zu finden, also einem Algorithmus welcher mit dem Input x versucht den Output y vorherzusagen. Also ist hier weniger das Informationsziel entscheidend, sondern vielmehr die Vorhersage gemessen an der Vorhersagegüte. Dieses Vorgehen trifft die Vorhersagen ohne Modellannahmen und kann somit flexibel zur Schätzung von Zusammenhängen eingesetzt werden.

Von ihm zunächst als Aufruf an die Statistiker gedacht, sich nicht nur rein auf die Datenmodellierung zu verlassen, haben Breimans Theorien der zwei Kulturen der statistischen Modellierung einige Diskussionen ausgelöst. So sagte zum Beispiel der Statistiker David Cox nach (Bre01): „Professor Breiman takes data as his starting point. I would prefer to start with an issue, a question or a scientific hypothesis.“. Er kritisiert also seine Modelle dahingehend, dass oft nicht reine Daten die Ausgangslage sind, sondern medizinische oder naturwissenschaftliche Fragen oder Hypothesen als Input vorliegen. Auch seine Veranschaulichung durch die Blackbox wird kritisch gewürdigt. So sagte Statistiker Bradley Efron nach (Bre01): „The whole point of science is to open up black boxes, understand their insides, and build better

boxes“. Nach Breimans Überlegungen gibt die „Algorithmic Modeling Culture“ Lösungsansätze für die neuartigen Probleme, die mit immer größer werdenden Datensätzen einhergehen. Diese Lösungsansätze des maschinellen Lernens machen eine reine Vorhersage, ohne einen Informationsgewinn dabei im Auge zu haben, sind deshalb aber auch flexibel. Brian Ripley sagte dabei auf der useR! Konferenz 2004 zum Unterschied zwischen der Statistik und dem maschinellen Lernen: „To paraphrase provocatively, 'machine learning is statistics minus any checking of models and assumptions'.“, siehe z.B. (Mik18). Somit beschreibt das erste Modell und damit die Statistik im Grunde genommen die Natur die hinter einem Vorgang, Daten oder einem Problem liegt. Für die Statistik bestehen Daten aus Zahlen. Machine Learning, also das zweite Modell, hilft bei reiner Prognose in Abhängigkeit ihrer Prognosegüte und Testfehler. Die Daten bestehen hier oft aus Bildern, Sprache oder sehr großen Datensätzen. Um diese Kluft wieder zu schließen sind sich aber viele Statistiker und Informatiker auch einig: „This comes back to education. If our students can't analyze giant datasets like millions of twitter feeds or millions of web pages then other people will analysze those data. We will end up with a small cut of the pie.“ (Was13). Wahrscheinlich ist Andrew Gelmans Antwort auf Ripleys Kommentar auch überspitzt gemeint: „In this case, maybe we should get rid of checking of models and assumptions more often. Then maybe we'd be able to solve some of the problems that the machine learning people can solve but we can't!“, siehe z.B. (Mik18). Jedoch wäre wohl eine Mischung der beiden letzten Kommentare ein guter Ansatz. Begonnen bei der Ausbildung von Studenten auf die aktuellen Anforderungen, wie etwa Big Data, in der realen Welt und der Wirtschaft, sollten flexible Methoden Einzug in die Statistik halten. So können sich beide Disziplinen, die Statistik und die KI, wie in der Vergangenheit gegenseitig positiv beeinflussen und voneinander profitieren.

Die Erste Welle der Künstlichen Intelligenz: Can machines think?

Zunächst folgt nun eine geschichtliche Einteilung beginnend mit der Künstlichen Intelligenz, um die einzelnen Aspekte besser zu verstehen. Dabei muss mit dem Turing-Test begonnen werden, dem Startschuss für die KI zur Zeit des 2. Weltkrieges. Alan Turing, ein britischer Logiker, Mathematiker und Kryptoanalytiker war während des Krieges an der Entschlüsselung von feindlichen Funksprüchen beteiligt und somit an der Entwickelung der ENIGMA, siehe auch Kapitel 2. 1950 schlug er den von ihm erarbeiteten Touring-Test für Maschinen vor und formte somit bis heute die noch unerreichte Zielsetzung für die KI. Laut seinem Artikel „Computing Machinery and Intelligence“ erschienen im Journal Mind im Oktober 1950 (Tur50) geht er seiner Frage „Can machines think?“ mit einem Experiment nach. Dieses nennt er das „Imitation Game“. Dazu setzt er zunächst zwei Personen in einen ersten Raum: Die männliche Person „A“ und die weibliche Person „B“. Eine dritte Person, „C“, stellt den Fragesteller dar und befindet sich in einem zweiten Raum. Ziel von Person „C“ ist es herauszufinden welche der beiden anderen Personen männlich beziehungsweise weiblich ist. So soll er am Ende des Experiments die Aussage: „A ist weiblich und B ist männlich“ oder eben „A ist männlich und B ist weiblich“ treffen können. Dies soll „C“ durch Fragen an „A“ und „B“ in schriftlicher Form erfahren. Person „A“ muss versuchen den Fragesteller in eine falsche Richtung zu lenken, Person „B“ hingegen soll helfen die richtige Lösung zu

finden. Die Frage in diesem Experiment ist nun wie oft eine Person „C“ die falsche Mutmaßung treffen wird. Turings Gedanke dahinter war, danach einen Schritt weiter zu gehen und Person „A“ durch eine Maschine zu ersetzen. Nun stellt sich die gleiche Frage: Wie oft wird ein Fragensteller falsch in der Annahme liegen, welche der Personen männlich und welche weiblich ist (Tur50)? Und diese Frage stellt sich natürlich erst wenn die Antworten der Maschine überzeugend genug sind, um den Fragesteller gleichwertig zu einem Menschen in die Irre führen zu können. So beschreibt er weiter, dass man einer solchen Maschine durch Programmierung die Fähigkeit geben muss so zu handeln wie es eine menschliche Person „A“ tun würde. Weiter ist eine zufällige Komponente bei der Programmierung wichtig um eine Art „freier Wille“ zu kreieren. Vereinfacht gesagt besteht eine Maschine also demnach den Turing-Test, falls eine Dritte Person, der Fragesteller, nicht erkennt welcher seiner beiden Gesprächspartner ein Mensch und welcher eine Maschine ist. Und dies unter den Bedingungen dass der Mensch und die Maschine sich in einem getrennten Raum ohne Sicht- und Hörkontakt zur Dritten Person befinden. Diese theoretischen Überlegungen zur Künstlichen Intelligenz wurden aber erst nach dem Tod Turings in der Dartmouth Konferenz 1956 ausgearbeitet. Somit gilt das Forschungsprojekt, ausgeschrieben Dartmouth Summer Research Project on Artificial Intelligence, als Geburtsstunde der künstlichen Intelligenz als akademisches Forschungsfeld. Zeitgleich mit der Einführung der ersten Programmiersprachen wie FORTRAN, zu lesen in Abschnitt 3.3, erreichte dieses Thema große Begeisterung und ein hoher Grad an Optimismus machte sich breit. Aus dem Förderantrag der Konferenz ist ins Deutsche übersetzt zu entnehmen, dass die Forschungsarbeiten des Projektes „auf der Grundlage der Vermutung durchgeführt werden, dass jeder Aspekt des Lernens oder jedes andere Merkmal der Intelligenz im Prinzip so genau beschrieben werden kann, dass eine Maschine zur Simulation geschaffen werden kann. Es wird versucht herauszufinden, wie man Maschinen dazu bringt, Sprache zu benutzen, Abstraktionen und Konzepte zu bilden, Probleme zu lösen, die heute dem Menschen vorbehalten sind, und sich zu verbessern.“. Weiter waren die Mitglieder der Konferenz „der Meinung, dass bei einem oder mehreren dieser Probleme ein bedeutender Fortschritt erzielt werden kann, wenn eine sorgfältig ausgewählte Gruppe von Wissenschaftlern einen Sommer lang gemeinsam daran arbeitet.“ (MMRS55). Bei den genannten Wissenschaftlern handelt es sich um John McCarthy vom Dartmouth College, Marvin Minsky von der Universität Harvard, Nathaniel Rochester von der I.B.M. Corporation, und Claude Shannon von den Bell Laboratories, neben einigen weiteren geladenen Teilnehmern. Die Themenpunkte der Konferenz sind „Automatic Computers“, „How Can a Computer be Programmed to Use a Language“, „[artificial] Neuron Nets”, „Theory of the Size of a Calculation” sowie „Randomness and Creativity” (MMRS55).

1965 programmierte der Informatiker Arthur Samuel in einem amerikanischen IBM-Labor einen Computer Dame zu spielen. Er erreichte dies mit einer Methode, die heute bekannt ist als das erste „selbstlernende“ Programm. Er verwendete dabei die sogenannte Alpha-Beta-Suche, oder auch Alpha-Beta-Pruning genannt. Dies erlaubt dem Programm einen Algorithmus, in dem es für den aktuellen Spielzug alle zukünftigen Spielzüge analysiert um die optimale Spielstrategie zu finden. Das Alpha-Beta-Pruning ist dabei eine effizientere Version des Minimax-Algorithmus. Diese Methoden funktionieren in einem Zwei-Personen-Nullsummenspiel mit perfekter und vollkommener Information. Das bedeutet es spielen zwei

Spieler gegeneinander, wobei nur einer gewinnen kann und dabei der andere verliert, und beide Spieler kennen den kompletten bisherigen Spielverlauf sowie die Regeln und alle möglichen Spielzüge (Har17). Diese Eigenschaften weisen die Spiele Schach, Dame oder Tic-Tac-Toe auf. Bei einer Betrachtung eines Entscheidungsbaumes stellt jeder Knoten k des Baumes einen möglichen Spielzustand dar. Je nach Ebene des Baumes wird derjenige Spieler dargestellt, der am Zug ist. Die jeweils möglichen zukünftigen Spielzüge werden durch ein Kind des aktuellen Zustandes dargestellt. Somit können vollständige Spielverläufe durch Betrachten des Entscheidungsbaumes von der Wurzel bis zum Blatt nachverfolgt werden, wobei d die Höhe des Baumes, also die vollständige Anzahl aller Spielzüge beschreibt. Somit enstehen k^d Blätter in der untersten Ebene des Entscheidungsbaumes. Um jeden möglichen Spielzug zu durchlaufen werden somit $T = \sum_{i=1}^{d} k^d$ Durchgänge des Algorithmus notwendig (Har17). Im Sinne des Minimax-Vorgehens gibt es in jeder Ebene ein anderes Spielziel. In der ersten Ebene, die die erste Runde des Spiels darstellt, wird das maximale Ergebnis gewählt, da der erste Spieler ja die Maximierung seines Spiels zum Ziel hat. In der zweiten Ebene wird aus den zur Verfügung stehenden Ergebnissen das Minimum gewählt, weil dieses Minimum ja wiederum das Maximum des Gegners darstellt. Dies wird iteriert, bis man am Ende des Entscheidungsbaumes angelangt und somit das Spiel beendet ist. Um nun diesen rechenaufwendigen Algorithmus mit T-Durchgängen zu beschleunigen, wird Pruning des Entscheidungsbaumes betrieben, eben das Alpha-Beta-Pruning. Dabei werden die Werte der möglichen Spielausgänge geschätzt und die Suche des Algorithmus auf der jeweiligen Ebene abgebrochen, sobald klar ist, dass kein weiterer Vorteil für den jeweiligen Spieler erzielt werden kann. α steht dabei für den größten bisher gefundenen Wert, β für den kleinsten Wert, welcher für das Spielziel der Runde genügt (Har17). Der „unnötige“ Zweig des Baumes wird gestutzt. So schrieb Samuel in seinem Artikel „Some Studies in Machine Learning Using the Game of Checkers“ übersetzt: “Die vielleicht elementarste Form des Lernens, die es wert ist, diskutiert zu werden, wäre eine Form des Routinelernens, bei der das Programm einfach alle während des Spiels vorkommenden Brettpositionen zusammen mit ihren berechneten Punktzahlen speichert. Dann könnte auf diesen Speicherdatensatz Bezug genommen werden und eine gewisse Rechenzeit eingespart werden. Dies kann kaum als eine sehr fortgeschrittene Form des Lernens bezeichnet werden, aber wenn das Programm dann die gesparte Zeit nutzt, um weiter zu rechnen, wird es sich mit der Zeit verbessern.” (Sam59). Somit setzte er einen ersten Meilenstein für die KI.

Ein weiterer folgte 1966 mit Joseph Weizenbaums ELIZA. Sie wird als die erste kommunikative Software bezeichnet und gilt als Wegbereiter für heutige Chatbots, Dialogsysteme mit sprachlichen Fähigkeiten zur Kommunikation über Instant-Messaging Systemen für einfache Sachverhalte, für die kein Mensch-Mensch Kontakt mehr benötigt wird. Unter anderem simulierte Weizenbaum ein Gespräch mit einem Therapeuten. Dessen natürliche Sprache simulierte er durch einen Thesaurus, eine Art Wörterbuch, das Relationen der beinhalteten Wörter innehat. So sucht ELIZA aus den Sätzen seines menschlichen Gesprächspartners Wörter heraus, zu denen Synonyme oder Oberbegriffe im Wortschatz gespeichert sind und gibt diese in Verbindung mit abgespeicherten Phrasen zu ähnlichen Themengebieten wieder. Einen Turing-Test hat der deutsch-amerikanische Weizenbaum jedoch mit ELIZA nicht bestanden, zu oft kam das Programm durch die Fragen der Testteilnehmer an seine Gren-

zen und wurde als Maschine entlarvt. Jedoch wurde mit der Wirtschaftskrise in den 1970er Jahren auch immer mehr die Fördermittel für dieses neue Feld der Forschung gekürzt. Auch konnten die hohen Erwartungen nicht erfüllt werden, was auch daran lag dass der Stand der Programmiersprachen und die Rechenleistung an ihre Grenzen kamen. Wie schon in Abschnitt 3.3 angeschnitten lag in dieser Zeit der Fokus auf der Weiterentwicklung und besseren Anpassung der Programmiersprachen. Dies erschaffte, wie zu lesen, eine bessere Basis für viele Fachbereiche der Statistik und somit auch für die KI (Cal19). Diese Phase wird jedoch als der erste KI-Winter bezeichnet. Die bereits erwähnten Simon und Newell versuchten schon seit 1956 einen General Problem Solver (GPS) zu entwickeln, also eine künstliche Intelligenz zu schaffen, die auf jedes Problemfeld eine Lösung bieten konnte. Ihr Programm nannten sie den "Logic Theorist", (NS56). Doch auch sie merkten im Zuge des Ersten KI-Winters, dass dies eine noch unmögliche Aufgabe darstellte.

Anfang der 1980er Jahre kamen KI-Wissenschaftler zu der Erkenntnis, dass man stattdessen mittlerweile durchaus leistungsfähige Expertensysteme aufbauen konnte. Dies beschreibt Computerprogramme die als Experte in einem bestimmten wissensintensiven Fachbereich fungieren. Dabei wird mit Hilfe von KI auf eine Wissensdatenbank eines bestimmten Gebietes mit Hilfe von Wenn-Dann-Regeln zurückgegriffen, um dem Benutzer sofortige Antworten und Problemlösungen auf Expertenniveau zukommen zu lassen. Begleitet durch die selbstständige Bereitstellung eines Lösungsweges, Schlussfolgerungen sowie der Möglichkeit des Lernens, also der Erweiterung und Verbesserung der Datenbank, siehe z.B. (Jac89). Diese Art der Wissensgewinnung und -konservierung war nun die erste Errungenschaft des Forschungsgebietes, um die KI die einen großen Mehrwert für Unternehmen und die Wirtschaft bedeutete. Sie halfen nun immer mehr bei der Finanzierung und hofften dadurch auf immer bessere Systeme. In dieser Zeit wurden auch einige weitere Programmiersprachen wie C++ oder MATLAB entwickelt und vor allem die Automatisierungstechnik wurde vorangetrieben. Doch schon bald stieß die KI-Forschung abermals an seine Grenzen und die angestrebten Ziele der Forschung wurden nicht erreicht. Die Wartung der entwickelten Maschinen wurde zu aufwendig und Desktop-Computer der Firmen Apple oder IBM waren inzwischen wesentlich leistungsstärker. Ein weiteres Mal war die Ungleichheit des Fortschritts zwischen der Hardware und der Software zu groß und der zweite KI-Winter begann, siehe z.B. (Cal19).

Die Zweite Welle: Vom Ensemble Learning zum Deep Learning

Aufgrund des abgeflachten Interesses während des zweiten Winters, konnte die KI-Forschung frei von der Kontrolle der Kapitalgeber ihrer Arbeit nachgehen. So entstanden bereichsübergreifende Synergien die einen großen Fortschritt für die Künstliche Intelligenz und besonders dem Machine Learning bedeuteten (Cal19). In der Mitte der 1990er Jahre wurde der Begriff des Random Forest hauptsächlich von dem bereits am Anfang des Kapitels erwähnten Leo Breiman geprägt. Random Forest ist ein Klassifikationsverfahren, in dem eine Reihe unkorrelierter Entscheidungsbäume randomisiert werden. Da maschinelles Lernen überwiegend durch Klassifizierung geschieht, wurde durch das innovative Random Forest die Trainingszeit verkürzt und so die Effizienz vor allem bei großen Datenmengen erhöht. Dabei gehört Random Forest zu den Bagging-Algorithmen, welche auch von Breiman in seinem Artikel

„Bagging Predictors“ 1996 im Magazin „Machine Learning“ publiziert wurden und seither das maschinelle Lernen revolutionierten. Dabei handelt es sich um Bootstrap-Algorithmen. Genauer um Bootstrap-Aggregationen, da sich auch der Begriff Bagging vom englischen ‚Bootstrap aggregating‘ ableiten lässt. Es werden dabei B Bootstrap-Stichproben vom Umfang n aus der Grundgesamtheit gezogen und eine Vorhersage m_i mit $i = 1, \ldots, B$ über das Modell getroffen. Für einen Wert x ergeben sich dann B Prädiktoren $m_i(x)$. Die einzelnen Bootstrap-Stichproben sind Replikationen des Originaldatensatzes, die dann in den Iterationen als neuer Lerndatensatz verwendet werden. Somit entstehen vielzählige Prädiktoren, welche daraufhin zu einem einzigen Prädiktor aggregiert werden. Dieser stellt den Durchschnittswert der Vorhersage einer Klasse mit Wert $m_B(x) = w_1 m_1(x) + ... + w_B m_B(x)$ dar. w_i mit $i = 1, ..., B$ ist die Gewichtung abhängig von der Modellgüte. Durch diese Methode können beim maschinellen Lernen erhebliche Genauigkeitssteigerungen erzielt werden. Außerdem wird die Komplexität von Modellen, speziell ihre Varianz, und somit overfitting (dt. Überanpassung) reduziert, siehe z.B. (Bre96).

Im Gegensatz dazu das Boosting, ein weiterer Maschine-Learning-Algorithmus. Dieser wurde auch in den 1990er Jahren von dem Informatiker und Professor an der Princeton-Universität, Robert Schapire, eingeführt. Im Vergleich zum Bagging erhöht es die Komplexität von Modellen mit hoher Verzerrung und vermeidet so underfitting (dt. Unteranpassung). Wie der Name schon sagt – aus dem englischen „Verstärkung“ – werden durch Kombination vieler schwacher Klassifikationsregeln bessere konstruiert, um so die genaueste Klassifikation vorzunehmen. Dabei wird der Prädiktor durch den Mittelwert beziehungsweiße der Mehrheit vieler Prädiktoren erstellt. Jedoch benutzt der Algorithmus hierzu die Residuen der vorherigen Iteration und gewichtet zudem die Entscheidungsregeln, um die stärksten hervorzuheben. Nach jeder weiteren Iteration werden die Gewichte angepasst und der Algorithmus „lernt“ daraus. So bezeichnet man den ersten Trainingsdatensatz als $b_1(x)$, schätzt und gewichtet ihn und verwendet die Residuen als neuen Trainingsdatensatz bis $b_M(x)$ mit $m = 1, \ldots, M$. Am Ende entsteht dann eine gewichtete Mehrheitsentscheidung für die Klasseneinteilung $b(x) = \text{sign}(\sum_{m=1}^{M} \alpha_m b_m(x))$ (Sch04).

Ein weiterer Algoritmus des Machine Learning wurde 1992 erstmals durch David Wolperts, in seinem Paper „Stacked generalisation“ erwähnt, das Stacking. Dies ist eine Methode bei der mehrere verschiedene Algorithmen, die sogenannten base-level-classifier oder level-0-classifier, kombiniert werden. So sollen wieder durch Gewichtung die Schwächen der Klassifizierer bei der Aggregation verringert und die Stärken hervorgehoben werden. Dabei hilft ein zusätzlicher Lernalgorithmus, der meta-classifier oder level-1-classifier, mit der Aufgabe zu lernen, welcher der Klassifikationsalgorithmen zu den besten Entscheidungen kommt. Entscheidend für den Erfolg des Stackings ist die richtige Wahl des level-1-classifier, siehe z.B. (Sch04).

Das Bagging wie auch das Boosting sowie Stacking lassen sich unter dem Begriff des Ensemble-Learning zusammenfassen. Denn bei diesen Algorithmen werden einzelne Prädiktoren zu einem Ensemble zusammengefasst, um so die optimale Lösung des Problems, hier der Klassifikation, zu finden. Das Prinzip basiert dabei prinzipiell auf dem Gesetz der großen Zahlen, siehe z.B. (Aun17). Für die Statistik wie auch das Machine Learning haben die Methoden des Ensemble-Learnings einen großen Fortschritt gebracht. Es konnten nun bessere

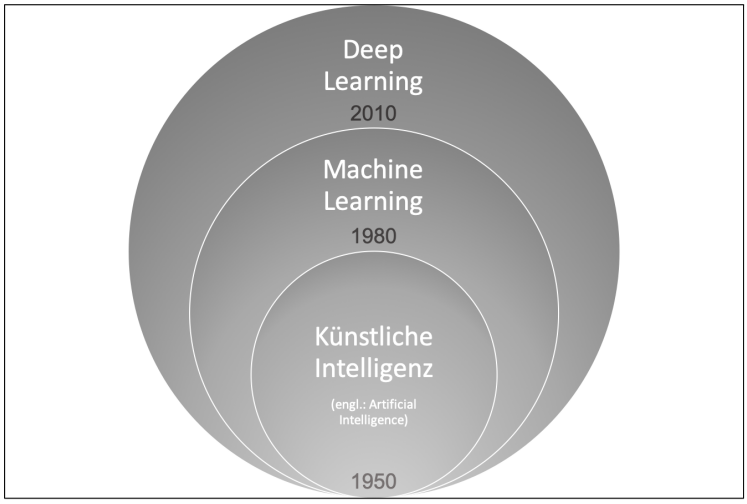

Abbildung 3.5: Von der Künstlichen Intelligenz zum Deep Learning; Eigene Darstellung

Modellergebnisse mit relativ wenig Rechenaufwand erreicht werden.

Zusammen mit der technischen Aufwärtsbewegung in den 90er und 2000er Jahren, und noch weiteren Programmiersprachen wie Phyton, Java oder auch dem in Abschnitt 3.3 aufgezeigten R wurden nun im zweiten Anlauf der KI bedeutende Meilensteine erreicht. Einer davon war der Sieg des von IBM auf das Spiel Schach programmierte Computer über den damaligen Schachweltmeister Garri Kasparow im Mai 1997, siehe z.B. (Kon). Aber auch für das immer brisanter werdendem Data Mining konnte die KI mittlerweile eine große Hilfe sein. Auch die Google-Suchmaschine hätte 1998 nicht ohne die erreichte Leistungsfähigkeit des Machine Learning in Betrieb genommen werden können, siehe z.B. (Cal19) und (Gö19). So muss man wie in Abbildung 3.5 zu erkennen ist, stets die KI als übergeordnete Disziplin betrachten, aus welcher sich die Teilbereiche, das Machine Learning und später das Deep Learning, entwickelt haben. Eine genaue Abgrenzung ist dabei natürlich nicht möglich.

Mittlerweile lässt sich das maschinelle Lernen in verschiedene Kategorien, basierend auf ihrer Lernfähigkeit, einteilen. Als Hauptkategorien gelten das supervised- und das unsupervised-learning, zu deutsch das überwachte und das unüberwachte Lernen. Beim supervised-learning werden Abbildungen vom Input x auf das Output y erlernt, also wie in der Abbildung 3.4 nach Breimans Algorithmic Modeling Culture. Dabei sind Beispielmodelle vordefiniert, um eine bessere Einteilung in die Klassen beziehungsweise Muster zu gewährleisten. Ein Beispiel hierzu ist die Klassifizierung von E-Mails, um so Spam-Mails von wichtigen zu unterscheiden. Auch bei der Gesichtserkennung beispielsweise in Social-Media wird es eingesetzt. Beim unsupervised-learning stehen nur Inputs x zur Verfügung. Es wird versucht interessierende Muster und Strukturen im Input zu erkennen, ohne zu wissen nach welcher Art Muster gesucht wird, eine Form der Wissensentdeckung. Dafür wird das Clustering betrieben, eine Einteilung in verschiedene Gruppen, oder die Hauptkomponentenanalyse, also die Reduzierung von Dimensionen. In einigen Quellen wird das teilüberwachte Lernen, eine Mischung der gerade erwähnten Methoden, oder das reinforcement-learning, zu deutsch das bestärkende Lernen, noch zur Hauptkategorie gezählt. Letzteres funktioniert über Belohnungs- und

Strafsignale, um das Lernen effizienter zu gestalten, siehe z.B. (Mur12).

Der neuste Zweig der KI ist das Deep Learning. Wie man der Abbildung 3.5 entnehmen kann, begann die Entstehung dieser Disziplin erst um 2010, wobei frühere Ansätze und Überlegungen schon seit Jahrzehnten in der Struktur der KI bestanden. Es wurden verschiedene Verarbeitungs- und Lernmethoden des menschlichen Gehirns als Vorbild genommen. Dabei kommen künstliche Neuronale Netze zum Einsatz. Die künstlichen Neuronen nehmen die Daten auf, bündeln, filtern und gewichten diesen Input und geben wiederum Daten aus. So entstehen mehrere Ebenen von Neuronen, welche das künstliche Neuronale Netz bilden. Diese haben, wie das natürliche Vorbild, Neuronen beziehungsweise Knoten als Vernetzungspunkte. Als Beispiel ist die Bilderkennung zu nennen. In der untersten Ebene werden vereinfacht gesagt Pixel eines Bildes den Neuronen zugeführt. Dort werden zunächst erste Helligkeitswerte erkannt. Als nächstes vielleicht Kanten und Ränder. Durch immer besseres Filtern und Verfeinern der erkannten Pixel ergeben sich in den weiteren Ebenen dann komplexere Muster. Je nachdem wie das Neuronale Netz im Vorfeld trainiert wurde, erkennt es dann beispielsweise Gesichter oder Tiere und kann sie in den weiteren Ebenen immer besser in Kategorien einordnen, wie weibliche oder männliche Merkmale eines Gesichtes oder der Unterscheidung zwischen einer Katze und einem Hund. Das Trainieren des Netzes und somit das Lernen erfolgt durch die schon erwähnten Lernmethoden, dem Überwachten-, Unüberwachten- und Bestärkenden-Lernen. Diese veranlassen die Anpassung der Gewichtung der einzelnen Neuronen, das Erstellung oder Löschung neuer Verbindungen zwischen Neuronen oder eine Anpassung der Schwellenwerte ab wann ein Neuron eine Ausgabe macht. Je mehr das System trainiert wird desto mehr Wissen hat es, um ein möglichst gutes Output zu approximieren. Im Deep-Learning werden starke Algorithmen durch mehrebige Neuronale Netze erstellt und trainiert, um komplexe Probleme zu lösen. Diese Probleme kommen beispielsweise bei der schon skizzierten Bilderkennung und -verarbeitung, bei Zeitreihenanalysen für Wetterprognosen, der Spracherkennung oder Frühwarnsystemen zum Einsatz. Diese noch junge Disziplin entspricht dem derzeitigen State of Art, also dem aktuellen Stand der Entwicklung des maschinellen Lernens. Siehe zu diesem Absatz z.B. (neu19).

Die Entwicklung der Künstlichen Intelligenz in den letzten 60 Jahren war rasant. Der Sprung von der Vorstellung denkender Maschinen von Turing oder Simon und Newell, die jedes Problem lösen können, hin zu hochkomplexen Algorithmen, die viele Bereiche des Lebens vereinfachen, wie die Unterstützung durch Sprachassistenten oder der sekundenschnellen Sprachübersetzung war gewaltig. Aber auch haben sie mittlerweile sehr wichtige Funktionen in der Medizin oder in der Früherkennung von Wetter- und Naturkatastrophen inne, siehe z.B. (Mel18) und (Med19). Doch wie weit ist die Entwicklung der KI derzeit? Chinesische Forscher haben 2017 eine Methode entwickelt, wie man Künstlichen Intelligenzen einen IQ-Wert zuordnen kann, um diesen mit dem Menschen zu vergleichen. Dabei wurden die Google-KI mit 47, oder der chinesischen Suchmaschine Baidu mit knapp 33 IQ Punkten verglichen. Dem Sprachassistenten Siri von Apple konnten nur ca. 24 Punkte zugewiesen werden. Als Vergleich mit dem durchschnittlichen IQ eines sechsjährigen Kindes von fast 56 Punkten liegen die derzeit eingesetzten Künstlichen Intelligenzen noch weit hinter der eines erwachsenen Menschen zurück (Bri17).

4 Eine Zitationsanalyse der Programmiersprache R

Dieser Teil der Arbeit beschäftigt sich mit einigen Fragen zur Entwicklung der Programmiersprache R. Nachdem im Abschnitt 3.3 die Anfänge der Programmiersprachen hin zum R Project und der Verwendung von R beschrieben wurde, soll dieses Kapitel die erwähnte Verbreitung und Anwendung von R verdeutlichen. Es wird eine Analyse der Entwicklung und der Verwendung von R betrieben, zum einen auf Basis der Länder, zum anderen der Fachbereiche in denen mit dem R Project statistische Programmierung durchgeführt wird.

4.1 Aufbau der Analyse

Die Zitationsanalyse ist ein Teilgebiet der Bibliometrie und wertet die Beziehungen zwischen zitierenden und zitierten Publikationen aus, um eine bestimmte Wissenschaftsentwicklung zu quantifizieren. Dabei werden bestimmte Aspekte wie das Fachgebiet, der Autor, der Ort, oder das Jahr einer zitierenden Publikation mit der Zitierten in Beziehung gestellt, siehe z.B. (Uni19). In diesem Fall wird als zitiertes Werk nicht ein bestimmtes Buch oder Paper verwendet, sondern die Programmiersprache R. Im Idealfall wird die Verwendung von R für die Erstellung einer Publikation von dessen Autor auch zitiert. Genau das wird sich hier zu Nutze gemacht. Als Zitationsdatenbank wurde die „Web of Science Core Collection“ verwendet, genauer dabei die „Cited Reference Search“. Die Suche fand auf Basis der Zitiervorschrift des R Projects statt. Diese ist unter der Kategorie „FAQ“ der CRAN Internetseite, siehe (Hor18) oder durch Aufruf von *citation()* mit R selbst zu finden:

> R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

So wurde in der „Cited Reference Search” als zitierten Autor *„R Core Team“* und als zitiertes Werk *„R: A Language and Environment for Statistical Computing“* verwendet. Um die Recherche besser zu strukturieren, wurden zu Beginn die Jahre *1990 – 2000* verwendet. Durch die stetig steigende Ergebnismenge wurde die Anzahl der gesuchten Jahre im weiteren Verlauf reduziert. Die zweite Suchanfrage wurde auf die Jahre *2001 – 2005* festgelegt, daraufhin *2006 – 2010* und ab 2011 für jedes Jahr einzeln. Im jeweils nächsten Schritt wurde nochmal überprüft, ob die Ergebnisse auch der Zitiervorschrift entsprachen. Diese variieren teilweise stark von der Zitiervorschrift des R Project, was zum einen natürlich auf verschiedene Versionen und Jahre zurückzuführen ist. Zum anderen aber auch bedingt durch Groß-

und Kleinschreibung, womögliche Tippfehler, durch Einflüsse anderer Sprachen oder auch schlicht falsches Zitieren. Jedoch wurden dabei auch jene Zitierweisen berücksichtigt, die den Punkten Autor und Werk der Zitiervorschrift ähneln und eine Zuweisung zum R Project erkennen ließen. Beispiele sind: „R-Core-Team“, „R: a language and environment for statistical computing“, “R CORE TEAM”. Nicht akzeptiert wurden Ergebnisse, die nichts mit der Sprache selbst zu tun hatten, also eindeutige falsche Suchergebnisse. Im nächsten Schritt wurden dann die Publikationen angezeigt, die R auf diese Weisen zitiert haben. Diese Publikationen wurden dann als „Full Record“, also dem vollen Datensatz zu jeder Publikation, exportiert und in R eingelesen. Für die Fragestellungen wurden die Datensätze der einzelnen Jahre zusammengefügt und die notwendigen Spalten bearbeitet. Diese sind unter anderem: „Autor“, „Titel“, „Sprache“, „Herausgeber“, also der Verlag, das „Publikationsjahr“, das „Land“ oder die „Kategorie“. Letztere stellt eine Einteilung durch das Web of Science dar, in welchen Fachbereich die Publikation einzuordnen ist. Zuletzt wurden geeignete Visualisierungsmöglichkeiten erstellt, um die Fragestellungen adäquat zu beantworten.

Ein weiterer Teil der Analyse ist die Betrachtung der Pakete die durch das CRAN bereitgestellt wurden. Die Pakete, mit kurzer Erläuterung und Datum der Veröffentlichung, wurden auch von der Internetseite des CRAN bezogen, siehe (pak19). Somit konnte in R eine Analyse auf Basis der Anzahl der veröffentlichten Pakete pro Jahr erstellt werden.

4.2 Fragestellungen

Die zu beantwortenden Fragestellungen sollen eine Veranschaulichung der schon im Abschnitt 3.3 aufgezeigten, steigenden Verbreitung von R darstellen. Auch sollen sie noch einen tieferen Einblick in die Entwicklung von R seit der ersten Veröffentlichung geben:

1. Wie hat sich die Anzahl der R Pakete über die Jahre verändert?
2. Wie haben sich die zitierenden Publikationen von R entwickelt?
   a) Auf Basis der Jahre
   b) Auf Basis der Fachbereiche
   c) Auf Basis der Länder in denen publiziert wurde

4.3 Ergebnisse

Fragestellung 1: Wie hat sich die Anzahl der R Pakete über die Jahre entwickelt?

In Abbildung 4.1 sind die Publikationsjahre auf der x-Achse zu der Anzahl der Pakete auf der y-Achse zu sehen. Der nahezu exponentielle Anstieg neu veröffentlichter Pakete durch das CRAN ist dabei zu erkennen. Wie in Abschnitt 3.3 erläutert, startete im Jahr 2000

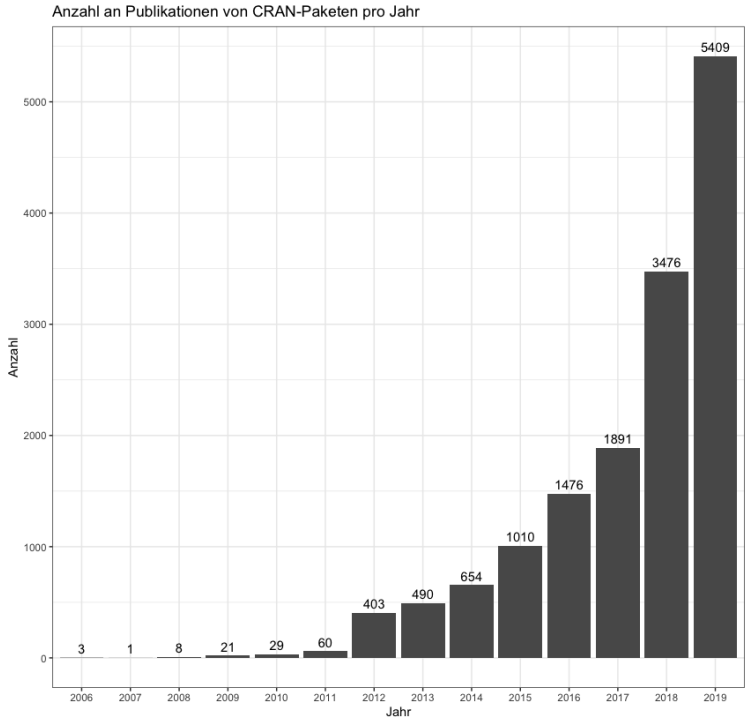

Abbildung 4.1: Veröffentlichte CRAN Pakete nach Jahren, Eigene Darstellung

die erste Version R 1.0.0 mit 12 Paketen. Sechs Jahre später wurden zunächst drei Pakete addiert, siehe (pak19). Diese waren: *„coxrobust“* zur Robusten Schätzung des Cox-Modells, *„BayesValidate“* zur Software-Validierung von bayesianischen Modellen, und *„allelic“* eine Testmethode. Im Jahr darauf lediglich nur ein Paket. Man erkennt einen ersten größeren Anstieg der Veröffentlichungsanzahl im Jahr 2012 auf 403 Pakete. In einer Ankündigung zur Veröffentlichung von R Version 2.15.0 werden „several new features and changes“, also einige neue Funktionen und Änderungen angekündigt, siehe (R2012). 2015 ist dann ein weiterer deutlicher Anstieg auf 1010 Pakete zu verzeichnen. 2018 wurden dann 3476 Pakete erstellt und zum Stand des Seitenaufrufs bereits 5409 im Jahr 2019. Zur Zeit der Analyse existieren insgesamt 14931 Pakete zuzüglich der 12 Pakete durch die erste Version.

Fragestellung 2: Wie haben sich die zitierenden Publikationen von R entwickelt?

Wie hat sich die Anzahl der zitierenden Publikationen von R über die Jahre entwickelt?

In Abbildung 4.2 zeigt sich die Häufigkeit einer Publikation mit Zitierung von R über die Jahre. Auch wenn R natürlich vor 2007 schon zitiert werden konnte, war es nicht möglich Publikationen vor diesem Jahr in der Web of Science Core Collection zu recherchieren. Mögliche Gründe hierfür könnten das Fehlen einer Zitiervorschrift vor dem Jahr 2007 sein, dass es erst seit einigen Jahren üblich beziehungsweise verbreitet ist, Software oder die benutzte

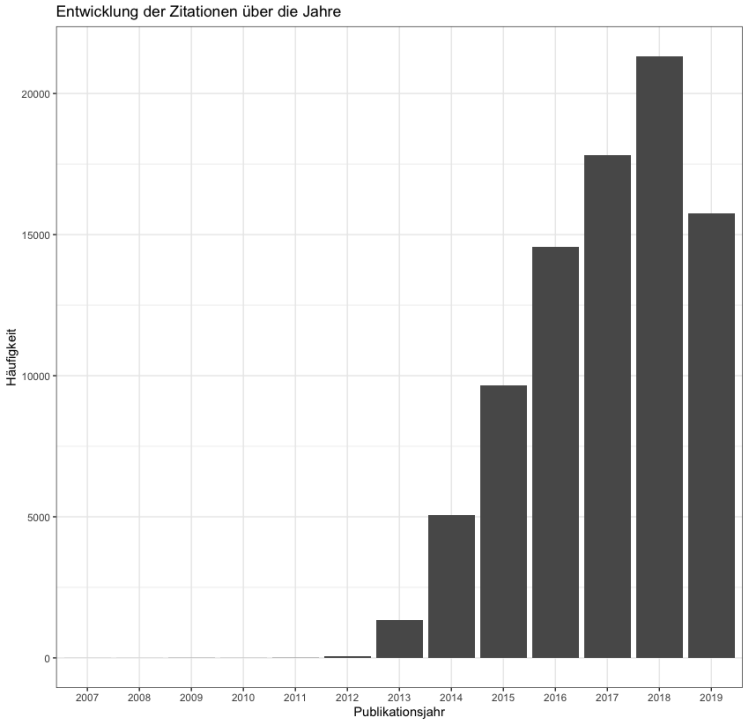

Abbildung 4.2: Häufigkeit der Zitation von R über die Jahre, Eigene Darstellung

Programmiersprache zu zitieren, oder die Web of Science Core Collection hat erst seit 2007 Zitationen für R vermerkt. In den Jahren 2007 bis 2011 gibt es nur einige wenige Publikationen im Vergleich zu den folgenden. 2012 sind bereits 71 vorhanden. In den weiteren Jahren ist ein Anstieg zu sehen. 2013 gibt es 1,351 Publikationen mit Zitierung von R, 2014 schon 5,064 und 2015, wie in der Abbildung zu erkennen ist, annähernd 10,000. Der bisherige Höchststand wurde 2018 mit über 21,000 Zitierungen von R. Wobei natürlich beachtet werden muss, dass noch keine endgültigen Zahlen für 2019 vorliegen können. Der Trend der vorherigen Jahre lässt aber einen weiteren Anstieg vermuten.

Wie hat sich die Anzahl der zitierenden Publikationen von R in den verschiedenen Fachbereichen entwickelt?

Eine Gesamtübersicht mit allen Publikationen in allen Fachbereichen über alle dokumentierten Jahre wird im Anhang bereitgestellt. Aufgrund der besseren Veranschaulichung werden hier nun die folgenden Ausführungen in Jahresintervalle geteilt.

Zunächst werden die in Fragestellung 1 erwähnten, schwächeren Jahre der Anzahl der Publikationen betrachtet, 2007 bis 2012. In Abbildung 4.3 zu erkennen sind die alphabetisch sortierten Kategorien der Fachbereiche der Publikationen auf der x-Achse. Diese Einteilung fand durch das Web of Science statt. Auf der y-Achse erkennt man die Häufigkeit der Publikationen im entsprechenden Fachbereich. Wie in der Legende zu erkennen sind die Balken jeweils eingefärbt in die Jahre in denen publiziert wurde. Dabei zu erkennen ist, dass die zwei Zitationen im Jahr 2007 in den Wissenschaftsbereichen „Ecology“, also zu deutsch der

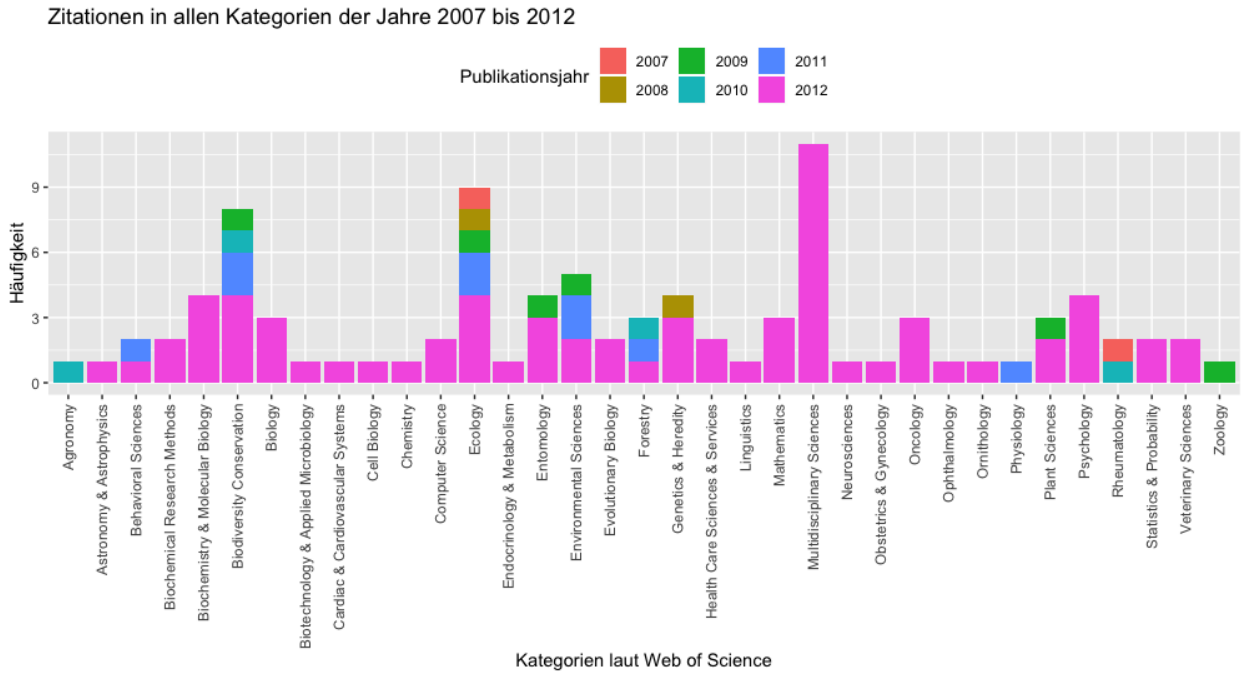

Abbildung 4.3: Fachbereiche der Zitationen von R der Jahre 2007 bis 2012, Eigene Darstellung

Ökologie und „Rheumatology“, der Rheumatologie stattfanden. Auch zwei Zitation hat das Jahr 2008 zu vermelden. Diese fanden zum einen auch in der Ökologie sowie in der „Genetics Heredity“, also der Genetik und Vererbung. 2009 dann gab es sechs Zitationen von R in den Fachbereichen Ökologie, Biodiversität, der Entomologie (Insektenkunde), der Umweltwissenschaften, Zoologie und Pflanzenwissenschaften. Zu erkennen ist, dass einige der Fachbereiche mit der Natur- oder Tierkunde zu tun haben. Die Ökologie ist aber nach wie vor auch vertreten. Im darauffolgenden Jahr 2010 kommen neben der Biodiversität und Rheumatologie die Kategorien „Forestry“, die Forstwirtschaft und „Agronomy“, zu deutsch die Agronomie oder Landwirtschaftslehre hinzu. Im Jahr 2011 wird R in den Fachbereichen der Verhaltenswissenschaften („Behavioral Sciences“), der Biodiversität, Ökologie, Umweltwissenschaften, Forstwirtschaft und „Physiology“, der Physiologie. Der Großteil der Zitationen in diesem Schaubild sind aus dem Jahr 2012, nämlich 71. Am häufigsten sind die „Multidisciplinary Sciences“ vertreten, also multidisziplinäre Wissenschaften. In diesem Jahr sind auch erstmals einige Fachbereiche erschienen, denen man der Oberkategorie Medizin und Biologie zuordnen kann, wie der „Biochemistry Molecular Biology“, der Biochemie und Molekularbiologie, Psychologie („Psychology“) oder „Health Care Sciences Services“, dem Gesundheitswesen um nur einige zu nennen. Auch sind nun die Mathematik („Mathematics“), die Statistik („Statistics Probability“) und die Informatik („Computer Science“) vertreten.

Aus Gründen der Übersichtlichkeit wurde das Diagramm zu den Jahren 2013 bis 2016 auf die häufigsten zehn Fachbereiche begrenzt. In der Anlage kann das Schaubild über die gesamt vertretenen Fachbereiche eingesehen werden. In Abbildung 4.4 sind die zehn häufigs-

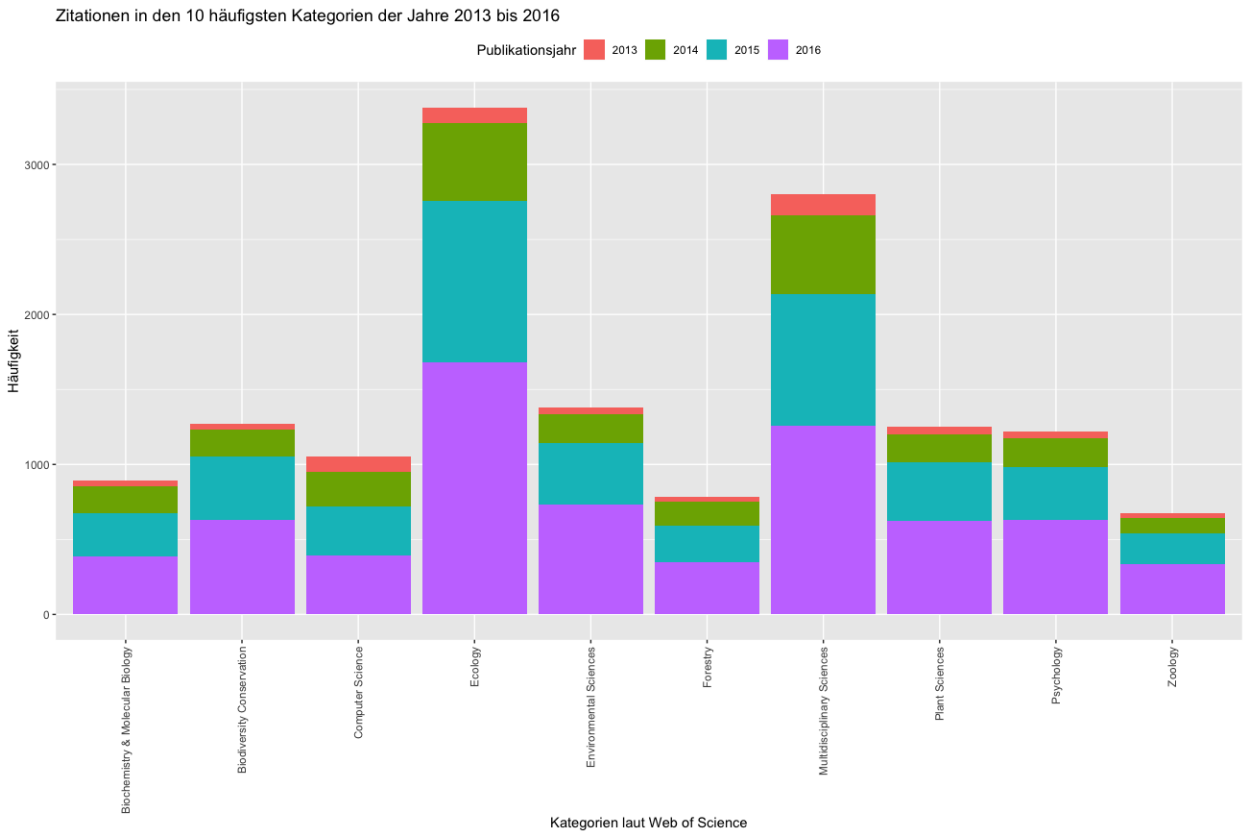

Abbildung 4.4: Die 10 häufigsten Fachbereiche der Zitationen von R der Jahre 2013 bis 2016, Eigene Darstellung

ten Fachbereiche der Jahre 2013 bis 2016 zu sehen. Wenn man dieses mit dem ersten Teil der zweiten Fragestellung, siehe Abschnitt 4.3 vergleicht, erkennt man den großen Anstieg an Zitationen der Programmiersprache und -umgebung R. Da seit 2013 viele Fachbereiche hinzugekommen sind, werden hier nur die zehn häufigsten Fachbereiche auf der x-Achse beschrieben. Die y-Achse stellt wieder die Häufigkeit der Zitation dar. Die über alle vier Jahre und auch in den Jahren 2015 und 2016 häufigste Kategorie stellt wieder die Ökologie dar. Die zweit häufigste sind multidisziplinäre Wissenschaften, welche auch 2013 und 2014 die meisten Zitationen, nämlich 141 und 527 aufweisen können. Diese hohe Anzahl kann man womöglich darauf zurückführen, dass dabei viele verschiedene Fachbereiche zusammenfasst wurden. Weiter sind die Informatik und Pflanzenwissenschaften ähnlich stark vertreten wie auch die Umweltwissenschaften. Auch unter den zehn häufigsten Fachbereichen der Publikationen sind Biochemie und Molekularbiologie, Biodiversität, Forstwissenschaft, die Pflanzenwissenschaften, Psychologie und die Zoologie. Insgesamt wird R 4,414 mal in diesen Jahren in Publikationen aus 176 verschiedenen Fachbereichen zitiert.

In Abbildung 4.5 ist ein Diagramm der 20 häufigsten Kategorien der Jahre 2017 bis 2019 zu sehen. Die zehn häufigsten dieser Jahre waren in der Verteilung der Fachbereiche denen der Jahre 2013 bis 2016 in Abbildung 4.4 sehr ähnlich. Deshalb stellt hier die x-Achse die

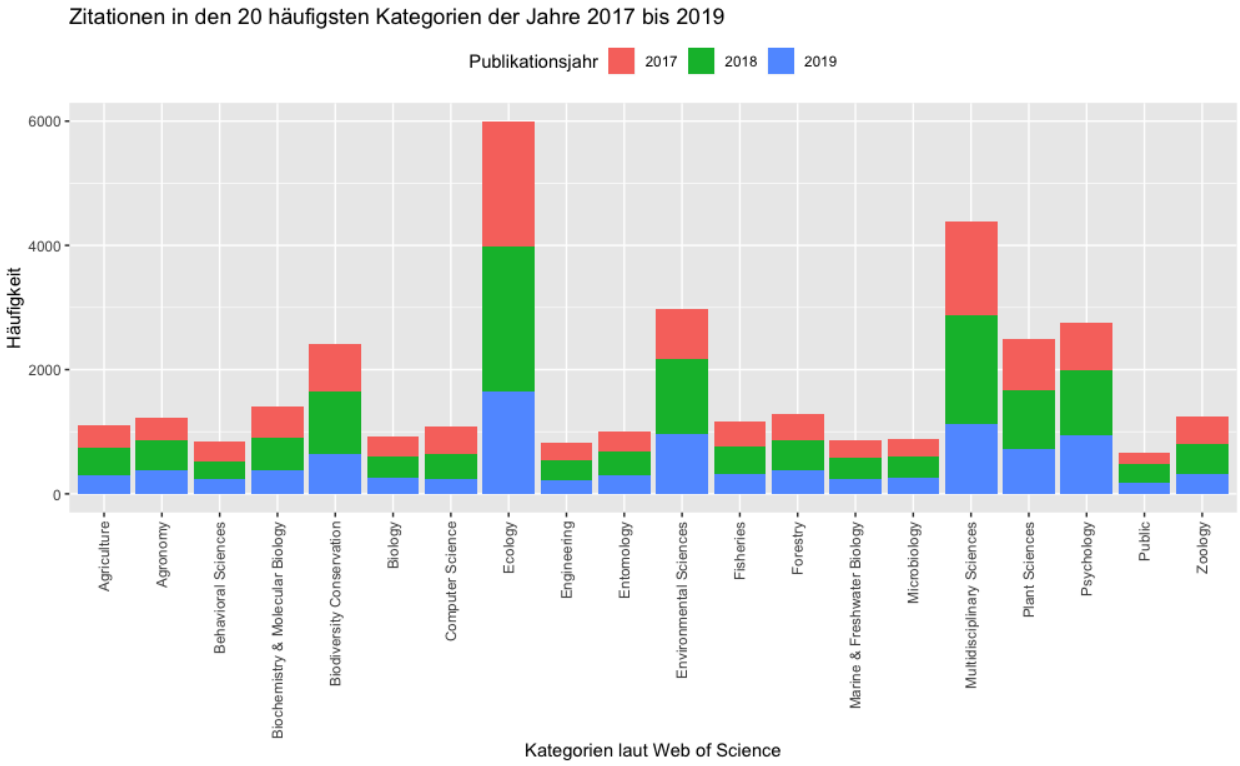

Abbildung 4.5: Die 20 häufigsten Fachbereiche der Zitationen von R der Jahre 2017 bis 2019, Eigene Darstellung

20 häufigsten Fachbereiche der Publikationen dar und die y-Achse wie gehabt die Häufigkeitsanzahl. Auch ist anzumerken, dass die Zahlen für das Jahr 2019 hier natürlich noch nicht vollständig sind. Wiederum stellt die Ökologie den größten Teil der Zitationen von R dar. Bisher in der Analyse noch nicht vorhanden war die 2013 erstmals erscheinende „Agriculture“. Diese unterscheidet sich von der bereits erwähnten „Agronomy“ dahingehend, dass ersteres die Wissenschaft der Landbewirtschaftung und Tierhaltung darstellt, zweiteres die Wissenschaft der Nutzung von Pflanzen, Tieren und Böden um daraus zum Beispiel Lebensmittel oder Brennstoffe herzustellen, siehe z.B. (agr). 2017 steht die Fischereiwirtschaft („Fisheries“) mit 391 Zitationen an 11. Stelle wie auch der sicherlich verwandte Fachbereich der Meeres- und Süßwasserbiologie („Marine Freshwater Biology“). Auch medizinische Fachbereiche sind nun immer mehr vertreten, wie die Klinische Neurologie („Clinical Neurology“), die Veterinärwissenschaft, die Medizin selbst, sowie auch die Parasitenforschung („Parasitology“), um nur einige Beispiele zu nennen. Das Ingenieurswesen („Engineering“) ist nun auch zu verzeichnen und kommt 2018 auf 316 Publikationen. Die Psychologie ist in allen drei Jahren stark vertreten und kommt auf 2,761 Zitierungen von R.

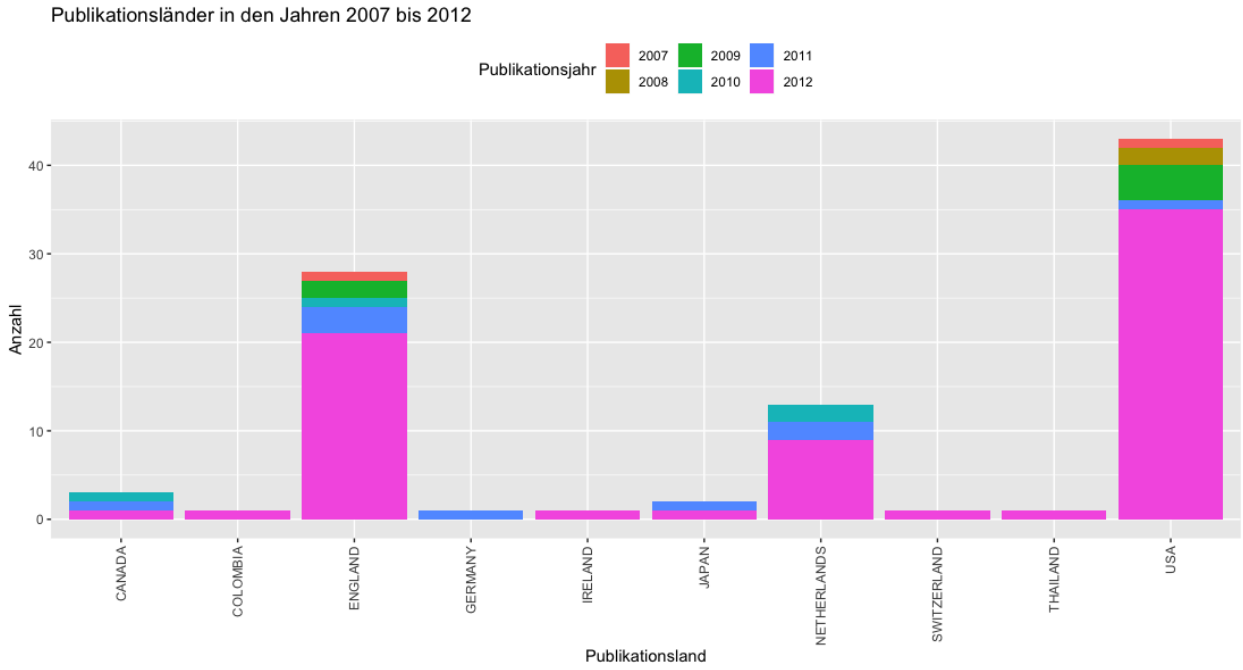

Abbildung 4.6: Die Publikationsländer der Zitationen von R der Jahre 2007 bis 2012, Eigene Darstellung

Wie hat sich die Anzahl der zitierenden Publikationen von R in den verschiedenen Ländern entwickelt?

Wie im vorherigen Abschnitt wird aufgrund der besseren Übersichtlichkeit in Jahresintervalle unterteilt. Auch hier liegt eine Gesamtübersicht aller Jahre und Länder dem Anhang bei. Zuerst werden wieder die Jahre 2007 bis 2012 in Abbildung 4.6 angezeigt. Hier zu sehen ist die Verteilung der Publikationen über die Länder in denen publiziert wurde. Diese sind alphabetisch geordnet auf der x-Achse abgetragen. Die y-Achse beschreibt die Anzahl an Publikationen. Wie schon in Abbildung 4.2 gezeigt, steigt die Anzahl pro Jahr mit jedem Jahr. Was man auch an der Farbverteilung der Balken, in der Legende mit dem jeweiligen Jahr gekennzeichnet, sehen kann. Am häufigsten wurde in den Jahren 2007 bis 2012 in den USA R in einer Publikation zitiert. Besonders 2012 mit 35 Zitationen in Publikationen sticht heraus. In England wurde R am zweithäufigsten zitiert. Insgesamt 28 mal und in 2012 davon 21 mal. In den Niederlanden 13 mal, jedoch tritt dies erst 2010 das erste Mal auf. Gefolgt von Kanada, Japan und jeweils eine Zitation in Kolumbien, Deutschland, Irland, der Schweiz und Thailand. Insgesamt sind 94 Publikationen in diesen Jahren zu finden.

Als nächstes werden die Jahre 2013 bis 2016 mit den zehn häufigsten Ländern in denen R zitiert wurde in Abbildung 4.7 gezeigt. Auf eine Gesamtübersicht wurde in dieser Arbeit aufgrund der Übersichtlichkeit verzichtet, diese kann aber wieder im Anhang eingesehen werden. Auf der x-Achse sind hier die häufigsten zehn Länder zu sehen in denen R in einer Publikation zitiert wurde und auf der y-Achse ist wieder deren Anzahl abgetragen. Es zeigt sich eine ähnliche Verteilung wie in der Abbildung 4.6 der Jahre 2007 bis 2012. Die USA

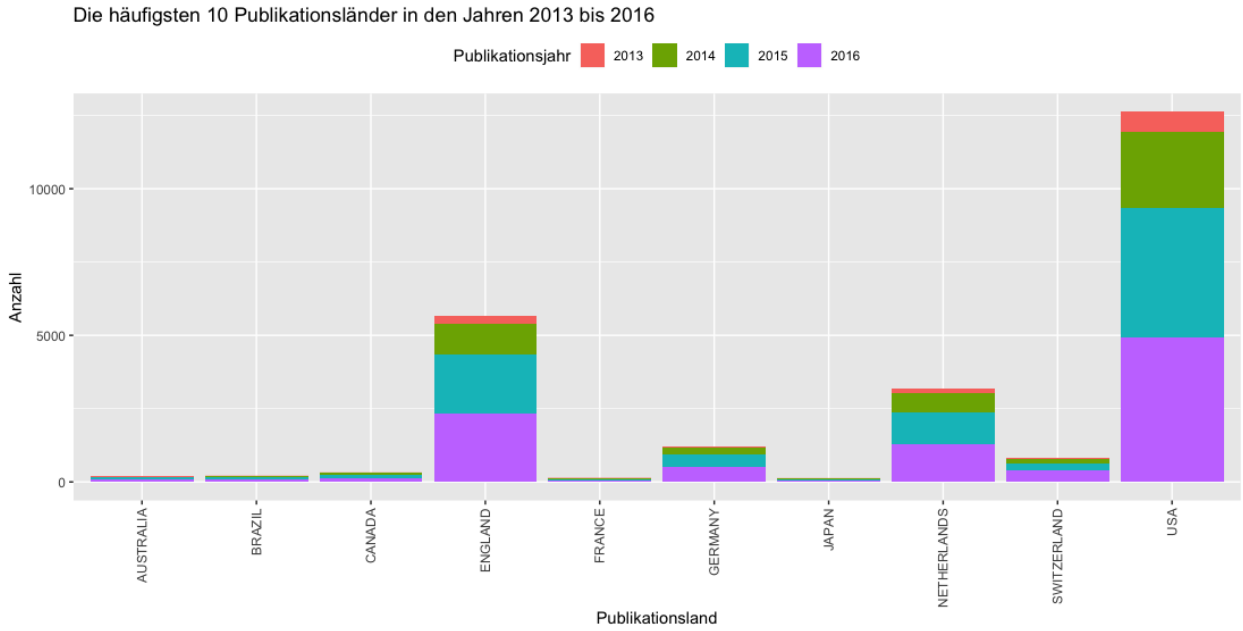

Abbildung 4.7: Die 10 häufigsten Publikationsländer der Zitationen von R der Jahre 2013 bis 2016, Eigene Darstellung

mit insgesamt 12,618 Zitationen, England und die Niederlande bilden die drei stärksten Länder, dieses Mal aber gefolgt von Deutschland mit hier insgesamt 1,222 Zitationen, der Schweiz und Kanada. Die Schlusslichter der Top-10 Verteilung bilden Brasilien, Australien, Frankreich und Japan.

Zuletzt werden in Abbildung 4.8 die Jahre 2017 bis 2019 abermals in Form der 10 häufigsten Länder in denen Publikationen mit R als Zitation veröffentlicht wurden. Dabei stellt die x-Achse wieder das jeweilige Publikationsland und die y-Achse die Anzahl dar. Die Verteilung der Länder ähnelt in diesen Jahren denen der vorherigen Abbildung 4.7 sehr, nur dass in den Jahren 2017 bis 2019 Irland statt Australien zu den Top zehn Ländern gehört. Die USA hat seit 2017 bis zum Datum der Datenerhebung 19,631 Zitationen publiziert. Im gesamten Betrachtungszeitraum sind es 32,292. Aus England kommen in den Jahren 2017 bis 2019 10,437 Zitationen und gesamt 16,135. Aus den Niederlanden kommen in den letzten drei Jahren 5,339 und insgesamt 8,541. Man sieht somit abermals die steigenden Zitationszahlen. Zusammengefasst wird R in 153 verschiedenen Ländern über 85,000 mal zitiert.

4.4 Fazit

Durch diese Analyse wurde die in Kapitel 3.3 beschriebene Verbreitung von R und dem R Project nochmal bestätigt. Es wurde die steigende Akzeptanz und Verwendung über die letzten Jahre aufgezeigt, welche nicht nur durch die stets gegebene Aktualität der Pakete und die damit verbundene Anpassung an den neuesten Stand der Statistik gegeben ist. Auch wurde gezeigt, dass R schon von Anfang an von nicht nur statistischen, mathematischen oder in-

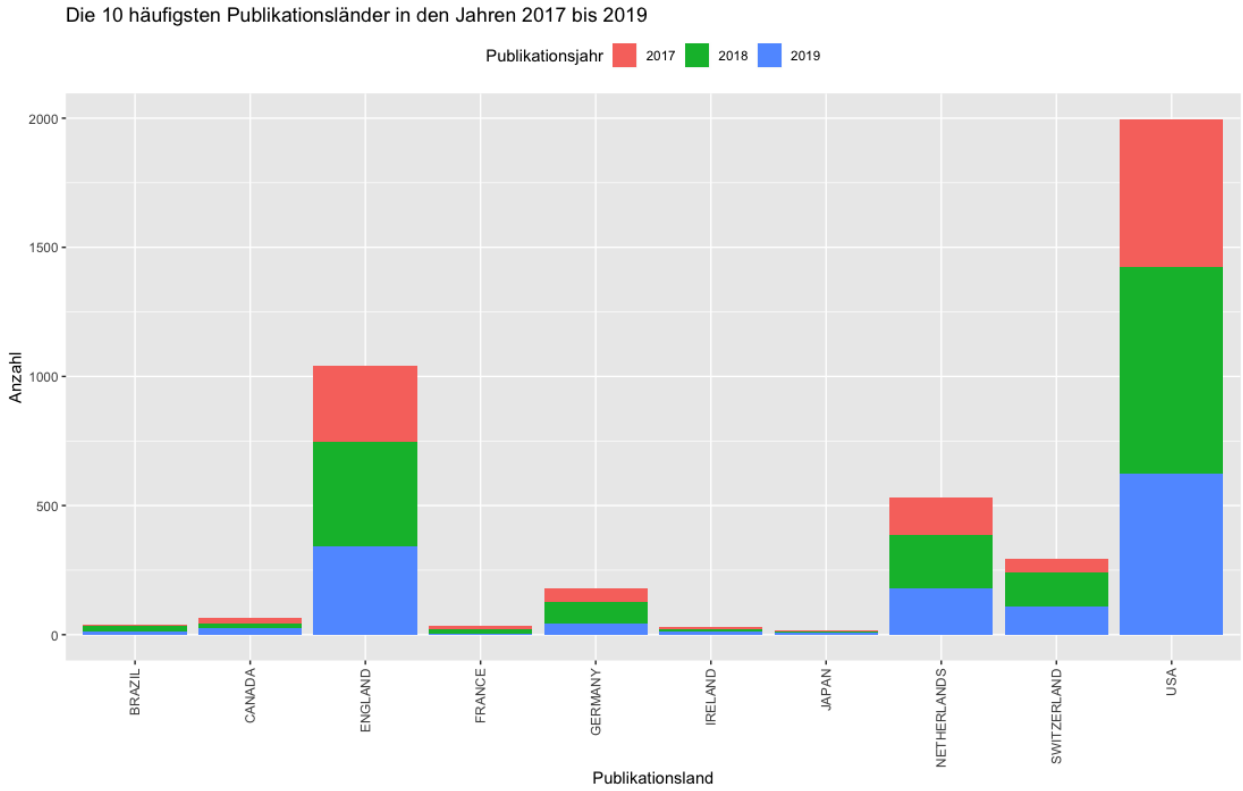

Abbildung 4.8: Die 10 häufigsten Publikationsländer der Zitationen von R der Jahre 2017 bis 2019, Eigene Darstellung

formatischen Wissenschaftsbereichen verwendet wurde, sondern besonders von der Ökologie, Biologischen Fachbereichen und einigen Unterkategorien der Landwirtschafts- und Forstwissenschaften. Auch Medizinische Bereiche wie die Psychologie, Rheumatologie und Neurologie verwendeten mit den Jahren immer öfter R für ihre Publikationen. Und das obwohl, wie in Abschnitt 3.3 erwähnt wurde, rein die Statistiksoftware SAS für medizinische Studien verwendet wird, da nur diese die Bestimmungen der amerikanischen Arzneimittelbehörde erfüllt. Begonnen hat die Verwendung, gemessen an der Zitation von R in Publikationen, vor allem in den USA, England und den Niederlanden. Diese Länder sind bis heute auch die mit den meisten Zitationen von R. Weiter stark vertreten sind Deutschland und Kanada. Doch viele andere Länder sind über die Jahre hinzugekommen, wie man den Grafiken in der Anlage entnehmen kann.

Diese Analyse stellt zwar keinen vollständigen Beweis und keine absoluten Zahlen über die Verbreitung und Verwendung von R dar. Jedoch ist dies ein Indiz dafür, dass R und das R Project ein Beispiel und somit auch eine Art Wegbereiter der computationalen Statistik ist. Nicht zuletzt dank der freien Verfügbarkeit von R und den Paketen ist es jedem möglich damit zu programmieren. Vor allem durch die Lehre von R an Universitäten werden Studenten der Statistik, Mathematik, Informatik, sowie verwandter Studiengänge, in der Programmierung mit R ausgebildet. Doch R endet nicht in der reinen Statistik selbst. Einige Pakete wie

"nnet", *"neuralnet"* oder *"caret"* sind auch für Machine-Learning Anwender interessant, denn mit ihnen können zum Beispiel künstliche Neuronale Netze trainiert werden, siehe (pak19). Man kann also gespannt sein, welche neuen Zweige der Statistik sich bilden und sich dabei gewiss sein, dass R stets „state of the art“ sein wird.

5 Schluss

Um nun abschließend die Ergebnisse und Erkenntnisse dieser Arbeit noch einmal zusammenzufassen und die Frage, wie sich die Methodik und die Anwendung in der Statistik durch die computationale Wende verändert hat, zu beantworten, werden nun zunächst die Ergebnisse des ersten Teils der Arbeit, der historischen Entwicklungen der Computertechnologien sowie der computationalen Statistik zusammengetragen. Daraufhin werden die Analyseergebnisse des zweiten Teils damit in Verbindung gebracht und erweitert.

Noch vor der computationalen Wende, als ein Computer als rechnender Mensch bezeichnet wurde, war die Statistik reine Buchhaltung oder die Bestimmung und Analyse von Volkszahlen. Doch in der Mitte des 20. Jahrhunderts fand eine erste Zusammenführung der Statistik mit dem Computer statt. Zunächst als Erleichterung von Berechnungen der, vor allem im Kriegseinsatz entstandenen, Rechenmaschinen, wurden diese bald auch in der Statistik eingesetzt. Denn die zu dieser Zeit entwickelten Markov-Chain und Monte-Carlo Verfahren fanden ihre Anwendung in rechenintensiven Zufallszahlengeneratoren, welche daraufhin in jedem Computer verbaut wurden. Auch Bayessche Methoden entfalteten sich erst durch die Nutzung einer Rechenmaschine. Als bis zu den 1990er Jahren Computer immer leistungsfähiger wurden, sowie der Heimcomputer seinen Durchbruch feierte, wurden die bisher eher theoretischen, statistische Methoden von den Computertechnologien endlich eingeholt und bisher unlösbare Probleme und Modelle konnten durch Simulationen gelöst werden. Viele der Anwendungen der Statistik wurden neu entdeckt und durch die Rechenleistung eines Computers nun letztendlich angewendet. Dies ergab auch neue Erkenntnisse und Möglichkeiten und es wurde bald klar, ein Statistiker muss sich mit Computer befassen. Auch in den 90er Jahren wurden R und das R Project begründet und als freie Programmiersprache und Software verbreitet. Durch die Programmiersprachen allgemein, welche schon 1954 mit der ersten höheren, wissenschaftlichen Sprache FORTRAN ihren Anfang fanden, fügte sich der Computer nun endgültig in die Anwendung der Statistik ein. Dies kam auch bald in den Universitäten und deren Lehre an. Wie der ehemalige Professor der Statistik an der Ludwig-Maximilians Universität München, Friedrich Leisch, in einem kurzen digitalen Interview die Ergebnisse dieser Arbeit bestätigte, wurden schon in den 90er Jahren spezielle Veranstaltungen wie das „Statistical Computing“ angeboten und zu Beginn der 2000 Jahre war R auch an der LMU als Fach der „Computerintensiven Methoden“ Pflicht. Durch R und die Möglichkeit der Paketerweiterungen und deren Anpassung an jedmögliche statistische Methode wurde die Anwendung der computationalen Statistik für alle frei zugänglich gemacht. Zufallszahlen, die noch vor wenigen Jahrzehnten eine große Hürde bei der Berechnung darstellten, können nun von jedem Statistik Studenten in Sekunden ausgegeben werden. Kein Wunder also, dass die Statistik nach immer mehr strebt und sich schon früh weitere Disziplinen daraus ergaben. Die Entwicklung der Künstliche Intelligenz, welche schon in den

1950er Jahren von Alan Turing sowie der Dartmouth-Konferenz ausgearbeitet wurde, war oft von kurzen Wintern durchbrochen, in denen die Grenzen der Computertechnologien erreicht wurden. Doch mit den immer leistungsfähigeren Machine-Learning Algorithmen der 1990er Jahre, der stetigen Weiterentwicklung des Computers und immer robusteren Programmiersprachen erreiche die Künstliche Intelligenz bedeutenden Meilensteine. Im zweiten Teil und der Zitationsanalyse von R als computationale Anwendung der Statistik, zeigt den Entwicklungsprozess der Programmiersprache und auch Programmierumgebung auf. Es wird dargelegt wie die Paketerweiterungen und somit die statistischen Anwendungsmöglichkeiten stetig augebaut und an die derzeitige Methodik angepasst wurden. Auch beweist die Analyse wie die Anzahl von Zitationen, und somit die Anwendung, der Programmiersprache in Publikationen stetig gewachsen ist. Nicht nur allgemein, sondern auch durch die Nutzung in verschiedensten Fachbereichen und Länder.

Um die Recherche und Analyse dieser Arbeit noch weiter auszuführen und darauf aufzubauen könnte man noch weiter in Richtung der Universitäten recherchieren. Für meinen Begriff ist ein Forschungsgegenstand erst dann in der Praxis angekommen, wenn Lehreinrichtungen diesen auch an die Schüler und Studenten weitergeben. Auch wäre eine noch detailliertere Auseinandersetzung mit der Zitationsanalyse der Programmiersprache R interessant. Zum einen wäre die Recherche vor dem Jahr 2007 notwendig. Dafür konnte in der Analyse des vierten Kapitels kein Ergebnis gefunden werden. Wenn der Grund für fehlende Zitationen in den Jahren davor an einer fehlenden Zitationsvorschrift von Software oder Programmiersprachen liegt, könnte man in anderen Datenbanken danach suchen, in Publikationen durch Lehrstühle oder Universitäten und Bibliotheken. Ein sicherlich weiterer, interessanter Aspekt wäre die Eingliederung von R in Unternehmen und deren Anwendung der Programmiersprache. Weiter fände ich eine Geschichte der computationalen Statistik am Institut der Statistik sehr aufschlussreich. Es könnten Zeitzeugen wie ehemalige und bestehende Statistik Professoren befragt werden und damalige Vorlesungsunterlagen untersucht und ausgewertet werden.

Wie diese Arbeit nun beweist hat die Statistik stets die computationalen Grenzen ausgereizt und gesprengt. Die neue Computertechnik hat stets die Statistik weitergebracht und in ihrer Methodik und Anwendung neue Möglichkeiten aufgezeigt. Doch auch die Statistik hat den Computertechnologien den nötigen Anreiz gegeben zu wachsen und gezeigt inwieweit sie sich entwickeln und verbessern müssen. Diese Symbiose und das Bestreben immer weitere Grenzen auszuloten und zu sprengen zeigt noch einmal die Richtigkeit von Bradley Efrons „Gesetz“ auf. Wer weiß welche neuen Erkenntnisse diese beiden Wissenschaften in den nächsten Jahren noch hervorbringen werden. Doch eines ist jetzt schon sicher, sie haben und werden sich stets zu neuen Herausforderungen motivieren. Dies ist nicht zuletzt der Grund für die herausragenden Geschehnisse der letzten 80 Jahre.

6 Anhang

Alle erwähnten, zur Analyse in Kapitel 4 gehörenden, Grafiken, Datensätze sowie der dazu erstellte R-Code sind in elektronischer Form bereitgestellt.

Abkürzungsverzeichnis

ASCC	Automatic Sequence Controlled Calculator
DERA	Darmstädter Elektronischer Rechenautomat
ENIAC	Electronic Numerical Integrator and Computer
UNIVAC	Universal Automatic Computer
MANIAC	Mathematical Analyzer Numerical Integrator And Computer Model
FORTRAN	Formula Translation
GUI	Graphical User Interface
TRADIC	Transistorized Airborne Digital Computer
WWW	World Wide Web
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
URI	Universal Resource Identifier
MCMC	Markov-Chain-Monte-Carlo
PRNG	Pseudo Random Number Generator
ML	Maximimum-Likelihood
GNU	GNU's Not Unix
GPL	General Public License
CRAN	Comprehensive R Archive Network
FSF	Free Software Foundation
KI	Künstliche Intelligenz
AI	Artificial Intelligence
GPS	General Problem Solver

Abbildungsverzeichnis

Literaturverzeichnis

[agr] *What is the difference between agronomy and agriculture?* https://bit.ly/2m2YkKZ. – Aufgerufen am: 21.09.2019

[Aun17] Aunkofer, Benjamin: Ensemble Learning. In: *Data science blog* (2017). https://bit.ly/2HlZtos. – Aufgerufen am: 09.09.2019

[Bre96] Breiman, Leo: Bagging predictors. In: *Machine Learning* 24 (1996), Aug, Nr. 2, 123–140. http://dx.doi.org/10.1007/BF00058655. – DOI 10.1007/BF00058655. – Aufgerufen am: 09.09.2019

[Bre01] Breiman, Leo: Statistical Modeling: The Two Cultures. In: *Statistical Science* 16 (2001), Nr. 3, 199-231. https://projecteuclid.org/download/pdf_1/euclid.ss/1009213726. – Aufgerufen am: 26.08.2019

[Bri17] Brien, Jörn: Google-KI doppelt so schlau wie Siri – aber ein Sechsjähriger schlägt beide. In: *t3n - digital pioneers* (2017). https://bit.ly/2khJ3Fq. – Aufgerufen am: 09.09.2019

[Bru18a] Bruderer, Herbert: *Band 1 Mechanische Rechenmaschinen, Rechenschieber, historische Automaten und wissenschaftliche Instrumente*. Auflage 2. Berlin, Boston : De Gruyter Oldenbourg, 2018 (Band 1). https://bit.ly/2k9krP4

[Bru18b] Bruderer, Herbert: *Band 2 Meilensteine der Rechentechnik - Erfindung des Computers, Elektronenrechner, Entwicklungen in Deutschland, England und der Schweiz*. Berlin, Boston : De Gruyter Oldenbourg, 2018 (Band 2). https://bit.ly/2mcTOJR

[Cal19] Calomme, Valentin: Die Geschichte der künstlichen Intelligenz. In: *ai blog* (2019). https://bit.ly/30JlHc0. – Aufgerufen am: 26.08.2019

[Chr00] Chrisley, Ronald ; Begeer, Sander (Hrsg.): *Artificial Intelligence: Critical Concepts*. Bd. 2. London, New York : Routledge - Taylor and Francis Group, 2000

[CT02] Christian Theis, Winfried K.: *Grundlagen der Monte Carlo Methoden*. Institut für theoretische Physik, Technische Universität Graz, Diss., 2002. https://itp.tugraz.at/MML/MonteCarlo/MCIntro.pdf. – Aufgerufen am: 06.08.2019

[dpa11] dpa: Chronik: Meilensteine der IBM-Geschichte. In: *Focus online* (2011). https://bit.ly/2lPmLeE. – Aufgerufen am: 09.09.2019

[DSC14] DSC: DSC 2014: Directions in Statistical Computing. In: *Directions in Statistical Computing* (2014). https://www.huber.embl.de/dsc/. – Aufgerufen am: 16.09.2019

[Efr00] Efron, Bradley: The bootstrap and modern statistics. In: *Journal of the American Statistical Association* (2000), S. 1293 – 1296. http://dx.doi.org/10.1080/01621459.2000.10474333. – DOI 10.1080/01621459.2000.10474333

[foc09] Vor 20 Jahren - Internet versus World Wide Web. In: *Focus online* (2009). https://bit.ly/2lWFAMQ. – Aufgerufen am: 15.09.2019

[Fre19] Free Software Foundation Inc.: Was ist GNU? In: *GNU Betriebssystem* (2019). https://www.gnu.org/home.de.html. – Aufgerufen am: 19.08.2019

[GG84] Geman, Stuart ; Geman, Donald: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. In: *Institute of electrical and electronics engineers - Transactions on Pattern Analysis and Machine Intelligence* PAMI-6 (1984), Nr. 6, 721–741. http://dx.doi.org/10.1109/tpami.1984.4767596. – DOI 10.1109/tpami.1984.4767596. – Aufgerufen am: 08.08.2019

[Gho18] Ghosh, Amit: Statistik-Software: R, Python, SAS, SPSS und STATA im Vergleich. In: *INWT Statistics* (2018). https://bit.ly/2TXdy0M. – Aufgerufen am: 23.08.2019

[Gru16] Gruber, Angela: Kleiner geht's nicht. In: *Spiegel online* (2016). https://bit.ly/2XiSVku. – Aufgerufen am: 16.09.2019

[Gr9] Grünwald, Robert: Wie man das optimale Statistikprogramm findet: Statistik Software im Überblick. In: *NOVUSTAT* (2019). https://bit.ly/2zkyPbn. – Aufgerufenam: 23.08.2019

[GS90] Gelfand, Alan E. ; Smith, Adrian F. M.: Sampling-Based Approaches to Calculating Marginal Densities. In: *Journal of the American Statistical Association* 85 (1990), Nr. 410, 398–409. http://dx.doi.org/10.1080/01621459.1990.10476213. – DOI 10.1080/01621459.1990.10476213. – Aufgerufen am: 08.08.2019

[Gö19] Göhrum, Christian: Die Geschichte von Google. In: *webwerkstatt* (2019). https://www.web-werkstatt.eu/die-geschichte-von-google/. – Aufgerufen am: 16.09.2019

[Har17] Hartmann, Maria: Der Alpha-Beta-Algorithmus. In: *Freie Universität Berlin* (2017). https://bit.ly/2kjYL2Z. – Aufgerufen am: 05.09.2019

[Hem19] Hemmerich, W. A.: Markov-Chain-Monte-Carlo-Verfahren — ohne Mathematik. In: *StatistickGuru* (2015 - 2019). https://bit.ly/2kocrtQ. – Aufgerufen am: 13.06.2019

[HK19] H. Klahr, C. M.: *Practical Numerical Training UKNum, Zufallszahlen, Monte Carlo Methoden*, Max Planck Institute for Astronomy, Heidelberg, Diss., 2018/2019. https://bit.ly/2koctls. – Aufgerufen am: 06.08.2019

[Hor18] Hornik, Kurt: *R FAQ - Frequently Asked Questions on R*. https://cran.r-project.org/doc/FAQ/R-FAQ.html#Citing-R. Version: 2018. – Aufgerufen am: 20.09.2019

[Igg18] Igges, Hans-Hermann: Vor 50 Jahren startete die Nixdorf Computer AG durch. In: *Neue Westfälische* (2018). https://bit.ly/2lQ6o1s. – Aufgerufen am: 15.09.2019

[Jac89] Jackson, Peter: *Expertensysteme. Eine Einführung*. Addison-Wesley Longman, Bonn, 1989

[Kli19] Kling, Bernd: Nvidia-CEO: Moore's Law ist am Ende. In: *ZDNet* (2019). https://bit.ly/2lXuM0Z. – Aufgerufen am: 16.09.2019

[Kna07] Knap, Michael: *Jackknife und Bootstrap*, Graz University of Technology, Institut für Theoretische Physik – Computational Physics, Diplomarbeit, Juni 2007. http://users.ph.tum.de/ga32pex/knap_bachelor.pdf. – Aufgerufen am: 13.08.2019

[Knu97] Knuth, Donald E.: *Art of Computer Programming, Volume 2: Seminumerical Algorithms (3. Auflage)*. Addison-Wesley Professional, 1997 https://bit.ly/2HJU3Gl. – Aufgerufen am: 06.08.2019

[Kon] Konitzer, Andreas: Geschichte des Computer. In: *Landesmedienzentrum Baden-Württemberg* https://bit.ly/2lXaOn5. – Aufgerufen am: 15.09.1991

[Les10] Leszczynski, Ulrike von: 100 Jahre Zuse - Der Computer, eine deutsche Erfindung. In: *Spiegel online* (2010). https://bit.ly/2kM9vXV. – Aufgerufen am: 09.09.2019

[Man18] Manhart, Klaus: Was Sie über Maschinelles Lernen wissen müssen. In: *Computerwoche - Voice of digital* (2018). https://bit.ly/2vlrrb4. – Aufgerufen am: 26.08.2019

[Med19] Was KI für die Medizin bedeutet. In: *Bundesministerium für Bildung und Forschung* (2019). https://bit.ly/2SCCl9N

[Mel18] Melanie: Die 5 wichtigsten Bereiche unseres Lebens, in denen künstliche Intelligenz eine wichtige Rolle spielen wird. In: *nine - cloud navigators* (2018). https://bit.ly/2kmE3iY. – Aufgerufen am: 16.09.2019

[MGNR12] Müller-Gronbach, Thomas ; Novak, Erich ; Ritter, Klaus: *Monte Carlo-Algorithmen*. Springer Berlin Heidelberg, 2012. http://dx.doi.org/10.1007/978-3-540-89141-3. http://dx.doi.org/10.1007/978-3-540-89141-3

[Mik18] Mike: The Two Cultures of Data Analysis. In: *Mike's STOR-i Blog* (2018). https://bit.ly/2kA3sG8. – Aufegrufen am: 13.09.2019

[MMRS55] McCarthy, John ; Minsky, Marvin ; Rochester, Nathaniel ; Shannon, Claude: A proposal for the dartmouth summer research project on artificial intelligence. In: *Formal Reasoning Group* (1955). http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html. – Aufgerufen am: 04.09.2019

[Moo65] Moore, Gordon E.: Cramming more components onto integrated circuits. In: *Electronics* 38 (1965), Nr. 8. https://intel.ly/2OZhsWY. – Aufgerufen am: 16.09.2019

[MRR+53] Metropolis, Nicholas ; Rosenbluth, Arianna W. ; Rosenbluth, Marshall N. ; Teller, Augusta H. ; Teller, Edward: Equation of State Calculations by Fast Computing Machines. In: *The Journal of Chemical Physics* 21 (1953), Nr. 6, 1087–1092. http://dx.doi.org/10.1063/1.1699114. – DOI 10.1063/1.1699114. – Aufgerufen am: 08.08.2019

[Mur12] Murphy, Kevin P.: *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012 https://bit.ly/2t6JrtI. – Aufgerufen am: 09.09.2019

[Neu51] Neumann, John von: Various techniques used in connection with random digits. In: *National Bureau of Standards Applied Mathematics Series, 12* (1951), S. 36–38. – Washington, D.C.: U.S. Government Printing Office

[Neu18] Neumann, Alexander: 25 Jahre: Wie R zur wichtigsten Programmiersprache für Statistiker wurde. In: *heise Developer* (2018). https://bit.ly/2HcViLW. – Aufgerufen am: 19.08.2019

[neu19] neuronalesnetz: Neuronale Netze - Eine EInführung. In: *www.neuronalesnetz.de* (2019). http://www.neuronalesnetz.de/downloads/neuronalesnetz_de.pdf. – Aufgerufen am: 15.09.2019

[NS56] Newell, Allen ; Simon, Herbert A.: The logic theory machine - A complex information processing system. In: *The RAND Corporation* (1956). https://bit.ly/2kO92V8. – Aufgerufen am: 16.09.2019

[Obs13] Obst, Ronert: *Computationale Wende, Paradigmenwechsel oder Sturm im Wasserglas?*, Ludwig-Maximilians-Universität München, Seminararbeit, 2013. – Aufgerufen am: 28.07.2019

[pak19] *Available CRAN Packages By Date of Publication*. https://cran.r-project.org/web/packages/available_packages_by_date.html. Version: 2019. – Aufgerufen am: 20.09.2019

[Pro16] Project, The Edsac R.: Tutorial Guide to the EDSAC Simulator. In: *The EDSAC Replica Project* (2016). https://www.dcs.warwick.ac.uk/~edsac/Software/EdsacTG.pdf. – Aufgerufen am: 15.09.2019

[Put10] Putt, Sarah: The story of R: a statistical tale with a twist. In: *Computerworld from IDG* (2010). https://www.computerworld.co.nz/article/489306/story_r_statistical_tale_twist/. – Aufgerufen am: 20.08.2019

[R C18] R Core Team ; R Foundation for Statistical Computing (Hrsg.): *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2018. https://www.R-project.org/

[R c19] R consortium: What is R Consortium. In: *R consortium* (2019). https://www.r-consortium.org. – Aufgerufen am: 31.05.2019

[R2012] *R 2.15.0 is released*. https://stat.ethz.ch/pipermail/r-announce/2012/000551.html. Version: 2012. – Aufgerufen am 20.09.2019

[RC11] Robert, Christian ; Casella, George: A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. In: *Statistical Science* 26 (2011), Nr. 1, 102–115. http://dx.doi.org/10.1214/10-sts351. – DOI 10.1214/10–sts351. – Aufgerufen am: 08.08.2019

[RDA04] Robert, Christian ; Doucet, Arnaud ; Andrieu, Christophe: Computational Advances for and from Bayesian Analysis. In: *Statistical Science* 19 (2004), Nr. 1, S. 118–127. http://dx.doi.org/10.1214/088342304000000071. – DOI 10.1214/088342304000000071

[Rip05] Ripley, Brian D.: How Computing has Changed Statistics. In: *Celebrating Statistics: Papers in Honour of Sir David Cox on His 80th Birthday* (2005), 197–211. http://www.stats.ox.ac.uk/~ripley/Cox80.pdf. – Hrsg. von A. C. Davison, Y. Dodge und N. Wermuth. Oxford University Press

[Rit96] Ritchie, Dennis M.: The Development of the C Language. In: *Bell Labs/Lucent Technologies* (1996). http://sites.harvard.edu/~lib113/reference/c/c_history.html. – Aufgerufen am: 19.08.2019

[Sam59] Samuel, Arthur L.: Some Studies in Machine Learning Using the Game of Checkers. In: *IBM Journal of Research and Development* 3 (1959), 210–229. https://bit.ly/2kCidZ3. – Aufgerufen am: 05.09.2019

[Sch04] Schwinger, Maximilian: Ensemble Models - Boosting, Bagging and Stacking. In: *Technische Universität München* (2004). https://bit.ly/1rWve7L. – Aufgerufen am 09.09.2019

[SdKS03] Schein, Edgar ; deLisi, Peter ; Kampas, Paul ; Sonduck, Michael: DEC is dead, long live DEC. In: *Time magazine* (2003), S. 38

[Sla17] Slawig, Thomas: Fortran im Wandel der Zeit. In: *heise Developer* (2017). https://bit.ly/33CRKvQ. – Aufgerufen am: 19.08.2019

[SN19] Statistik-Nachhilfe: Bootstrapping. In: *Ratgeber* (2019). https://bit.ly/2k9KvcW. – Aufgerufen am: 12.08.2019

[Soa16] Soare, Robert I.: *Turing computability: theory and applications*. Berlin, Heidelberg : Springer-Verlag, 2016

[Ste11] Steuer, Detlef: Simulation und Prognose. In: *Fakultät Wirtschafts- und Sozialwissenschaften, Helmut Schmidt Universität* (2011). http://fawn.hsu-hh.de/~steuer/downloads/FT2010/SP-11.pdf. – Aufgerufen am: 16.09.2019

[Ste15] Stein, Florian: Social-Media-Entwicklung & -Geschichte im Überblick in Deutschland. In: *social-media-agentur.net* (2015). https://bit.ly/2DfunxB. – Aufgerufen am: 16.09.2019

[The19] The R Foundation: What is R? In: *The R Project for Statistical Computing* (2019). https://www.r-project.org/about.html. – Aufgerufen am: 31.05.2019

[Tod18] Todesco, Rolf: ENIAC. In: *hyperkommunikation* (2018). https://www.hyperkommunikation.ch/lexikon/eniac.htm. – Aufgerufen am: 14.09.2019

[Tuk62] Tukey, John W.: The Future of Data Analysis. In: *The Annals of Mathematical Statistics* 33 (1962), Nr. 1, 1–67. http://dx.doi.org/10.1214/aoms/1177704711. – DOI 10.1214/aoms/1177704711

[Tuk65] Tukey, John W.: The Technical Tools of Statistics. In: *The American Statistician* 19 (1965), Nr. 2, 23. http://dx.doi.org/10.2307/2682374. – DOI 10.2307/2682374. – Aufgerufen am: 13.08.2019

[Tur36] Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem. In: *Proceedings of the London Mathematical Society* s2-42 (1936), Nr. 1, 230 – 265. https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf. – Aufgerufen am: 14.09.2019

[Tur50] Turing, A. M.: Computing machinery and intelligence. In: *Mind* LIX (1950), Nr. 236, 433–460. http://dx.doi.org/10.1093/mind/LIX.236.433. – DOI 10.1093/mind/LIX.236.433. – Aufgerufen am: 04.09.2019

[Uni19] Universität Ulm: Zitationsanalyse - Bibliometrie - Infometrie. In: *Kommunikations- und Informationszentrum (kiz)* (2019). https://bit.ly/2m1zvin. – Aufgerufen am: 20.09.2019

[Wag11] Wagner, Helga: *Einführung in die Bayes-Statistik: Markov Chain Monte Carlo Verfahren*. AG Method. Grundlagen der Statistik & ihre Anwendungen, Ludwig-Maximilians-Universität München, Institut für Statistik, Diss., WS2010/11. http://thomas.userweb.mwn.de/Lehre/wise1011/Bayes_1011/. – Aufgerufen am: 06.08.2019

[Was13] Wasserman, Larry: Rise of the Machines. In: *Department Data Archive* (2013). https://www.stat.cmu.edu/~larry/Wasserman.pdf. – Aufgerufen am: 27.08.2019

[Wat11] Watnik, Mitchell: Early Computational Statistics. In: *Journal of Computational and Graphical Statistics* 20 (2011), Nr. 4, 811–817. http://dx.doi.org/10.1198/jcgs.2011.204b. – DOI 10.1198/jcgs.2011.204b. – Aufgerufen am: 21.05.2019

[YR86] Yang, M. ; Robinson, D.: *Understanding and Learning Statistics by Computer*. World Scientific, 1986. http://dx.doi.org/10.1142/0213. http://dx.doi.org/10.1142/0213. – Aufgerufen am: 13.08.2019

Aus optischen Gründen wurden lange URL mit dem URL-Kürzungstool auf https://bitly.com gekürzt

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum Unterschrift