http://fc06.deviantart.net/fs71/f/2011/043/3/9/chaos_tree_by_beronheavyhand-d39edmy.png

Parameterwahl für die Populationssynthese mittels Regressionsbäumen

Yannick Pfeifhofer

Supervisors:
Prof. Dr. Kay W. Axhausen
Kirill Müller

Semesterarbeit
Studiengang Bauingenieurwissenschaften **Juni 2014**

Institut für Verkehrsplanung und Transportsysteme
Institute for Transport Planning and Systems

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Einführende Begriffe

Attribute:	Eigenschaften, durch die reale Personen beschrieben werden
Agent :	Abbildung der Attributenkombination von realen Personen
Synthetische Population :	Population von Agenten
Observation:	Agent oder Person in der entsprechenden Population
Stichprobe:	Teilmenge der realen Population
Multivariate Wahrscheinlichkeitsverteilung:	Funktion, die jeder Attributenkombination eine Auftretenswahrscheinlichkeit zuweist

Bezeichnungen

$a_{i,j}$:	Attribut i in der Ausprägung j
A_i:	Kontrollvariable
f:	Expansionsfaktor
p:	Wahrscheinlichkeit

Bachelor Arbeit

Parameterwahl für die Populationssynthese mittels Regressionsbäumen

Yannick Pfeifhofer

IVT-ETHZ

Telefon: 076 6798 093

yannickp@student.ethz.ch

Juni 2014

Kurzfassung

Agentenbasierte Mikrosimulationsmodelle werden zurzeit als bester Ansatz angesehen, um komplexe, dynamische Verkehrsmodelle zu erzeugen. Diese werden verwendet, um die zeitliche Evolution eines Systems (hier des Verkehrssystems) simulieren zu können, sodass sich die räumliche Verteilung aller Teilnehmer zu einem bestimmten Zeitpunkt in der Zukunft so präzise wie möglich darstellen lässt. Dies wird ausgenutzt, um die Verkehrsnachfrage abzuschätzen und zu modellieren. Um dies zu ermöglichen, muss als erstes eine Population von Agenten (die als Verkehrsteilnehmer fungieren) synthetisiert werden, die die reale Population am besten repräsentiert. Diese künstliche Population wird dann als Eingabegrösse für das Modell verwendet. Die Herausforderung besteht somit darin, mit den wenigen, zur Verfügung stehenden Informationen einen kompletten Datensatz zu erzeugen, der alle Agenten samt ihrer Eigenschaften, den Attributen, enthält und als äquivalent zur realen Population betrachtet werden kann. Dieser Datensatz wird mittels der zu bestimmenden, multivariaten Wahrscheinlichkeitsverteilung erstellt. Diese Verteilung bestimmt die Zusammenstellung der synthetischen Population, indem sie allen Attributenkombinationen eine Auftrittswahrscheinlichkeit zuweist, so dass bestimmt werden kann, welche Agententypen und in welcher Anzahl in die Population miteinbezogen werden.

Diese Arbeit ist in zwei Teile gegliedert. Der erste Teil befasst sich mit den wichtigsten, in der Literatur gefundenen Techniken der Populationssynthese und der zweite mit einem neuen, in R implementierten und auf CART-Regressionsbäumen aufbauendem Verfahren, welches eingeführt, beschrieben und einer Parameterstudie unterzogen wird.

Schlagworte

Mikrosimulation, Populationssynthese, Regressionsbäume, Attribute, multivariate Wahrscheinlichkeitsverteilung.

Zitierungsvorschlag:

Y. Pfeifhofer (2014) Parameterwahl für die Populationssynthese mittels Regressionsbäumen, *Bachelorarbeit Studiengang Bauingenieurwissenschaften*, Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich, Zürich.

1 Literaturrecherche

1.1 Problemstellung und Einleitung

Mit der Populationssynthese wird das Ziel verfolgt, einen kompletten Datensatz des zu untersuchenden Gebietes zu erzeugen, der alle Personen samt ihrer Eigenschaften enthält. Dies wird erreicht, indem alle diese Personen auf synthetische Agenten abgebildet werden, die durch deren Attributenkombination definiert sind und die synthetische Population bilden. Im Allgemeinen kann eine solche künstliche Population nicht direkt auf Basis der zur Verfügung stehenden Daten konstruiert werden, weil diese nie in ausreichender Menge und Form abrufbar sind. Da die Anonymität jeder betroffenen Person der Stichprobe gewährleistet sein muss, muss man sich generell mit kleinen, aufgeschlüsselten Stichproben, auch als Mikrostichproben bekannt, und einer Reihe von aggregierten begnügen, von denen dann der komplette, erforderliche Datensatz sinnvoll abgeleitet wird. Unter aufgeschlüsselter Stichprobe ist ein Datensatz zu verstehen, der eine Menge von Agenten samt ihrer kompletten Beschreibung und somit die Korrelationsstruktur der Attribute in der Population enthält. Aggregierte Stichproben hingegen sind Datensätze, in denen nur die aufsummierten Randhäufigkeiten eines oder mehrerer korrelierter Attribute enthalten sind, aus denen aber die gesamte Korrelationsstruktur der Population nicht abgeleitet werden kann. Mikrostichproben stammen meistens aus einer viel grösseren Region als der untersuchten, während sich die aggregierten Stichproben auf eine bedeutend grössere Populationsmenge beziehen und einer kleineren Region angehören, also eine schärfere Auflösung besitzen. Beispiele der Mikroproben sind die aus den USA stammenden PUMS (Public-Use Microdata Samples), die kanadischen PUMF (Public-Use Microdata Files), und das Sample of Anonymized Record aus dem U.K. Beispiele der aggregierten Stichproben sind die Summary Files aus den USA, die Basic Summary Tabulations aus Kanada und die Small Area Statistics aus dem U.K. Da die zwei Datensätze aus verschiedenen Quellen stammen können, beziehen sie sich nicht ausschliesslich auf dieselbe Agentenmenge. Diese Inkonsistenz kann zu einer Reihe von Problemen führen, auf die später eingegangen wird. Alle Agenten einer Population werden eindeutig mittels ihrer Eigenschaften beschrieben, die in Attribute unterteilt werden. Diese können nominal oder ordinal sein. Alle Attribute werden ihrerseits noch in verschiedene Kategorien unterteilt. Jeder Agent kann somit mittels einer kodierten Attributenkombination eindeutig beschrieben werden. Alle Agenten mit derselben Kombination werden einer Gruppe zugewiesen. Alle Gruppen bestehen somit aus einer einzigartigen Attributenkombination A $\in \{a_{1,k}, a_{2,l}, \dots, a_{n,z}\}$ von n Attributen $a_{i,j}$. Dabei bedeuten die Parameter i i=1, … ,n die n verschiedenen Attributentypen und j=1,…,$k/l/z$ die Anzahl an Kategorien, in die jedes Attribut gespalten wird und somit die verschiedenen Ausprägungen, in denen dieses Attribut auftritt.

Um das Problem des Datenmangels erfolgreich lösen zu können, sind eine Vielzahl von mathematisch-statistischen Methoden entwickelt worden, die es trotzdem ermöglichen, eine synthetische Population von hoher Qualität zu erzeugen. Dies geschieht im Allgemeinen über die Zusammenführung der beiden vorab beschriebenen Datenarten, mit dem Ziel der Beibehaltung der Korrelationsstruktur aus der Mikroprobe und deren Verbesserung mittels der Anpassung an die Randverteilungen. Auf diese Weise wird eine realistische, multivariate Verteilung der Attribute abgeleitet, die man dann zur Erzeugung der erforderlichen Anzahl von synthetischen Agenten verwendet. Dies geschieht mittels einer Zuweisung von Attributenkombinationen in Form von Punkten in den endlichen Teilraum des $\mathbb{Z}^d$, der somit als Zielmenge der Verteilung dient und in eine Datentabelle rücktransformiert werden kann. Die Hauptaufgabe jedes Synthetisierungs-Algorithmus besteht darin, diese Verteilung in optimaler Weise mit den zur Verfügung stehenden Daten abzuschätzen. Dieser Prozess, der der

Erstellung einer synthetischen Population dient, ist unter dem Namen Populationssynthese bekannt.

1.2 Techniken zur Populationssynthese

Hauptsächlich sind in der Literatur zwei ältere Familien von Techniken zu finden: die synthetischen Rekonstruktionen und ihre Variationen und die kombinatorischen Optimierungen. Diese Anpassungsverfahren leiten die multivariate Wahrscheinlichkeits-funktion der Attribute von einer auf die aggregierte Stichprobe angepassten, aufgeschlüsselten Datentabelle ab. Mittels dieser Verteilung wird dann die Mikroprobe zur synthetischen Population expandiert. Die neueren „imputation"-Verfahren (Farooq *et al.* (2013)) hingegen brauchen nicht zwangsläufig beide Stichproben, aber auch sie verfolgen das Ziel, die multivariate Verteilung in optimaler Weise abzuschätzen. Der grösste Unterschied zu den älteren Techniken besteht darin, dass die Agenten direkt mit dieser Verteilung erzeugt werden. Dies bringt eine Reihe von Vorteilen mit sich. Der erste Schritt aber ist für alle Verfahren gleich und besteht darin, die für relevant gehaltenen, soziodemographischen Attribute festzulegen, damit jede Gruppe $A_i = \{a_1; a_2; \dots; a_n\}$ definiert werden kann. Jeder Agent kann somit einer Gruppe zugewiesen werden, die aus einer einzigartigen Kombination von Attributen a_i besteht. Jeder zu erzeugende Agent kann als Realisation des Zufallsvariablenvektors $\{A_1, A_2, \dots, A_n\}$ gesehen werden, mit der Wahrscheinlichkeit, die von der zu findenden, multivariaten Verteilungsfunktion vorgegeben wird.

1.3 Synthetische Rekonstruktionen

Diese Methoden vereinen die zwei unterschiedlichen, aggregierten und aufgeschlüsselten Datenquellen, um die multivariate Verteilungsfunktion der Attribute der realen Population abzuschätzen. Das passiert mittels einer Anpassung der aufgeschlüsselten Stichprobe, so dass die marginalen Randverteilungen oder Kontrollvariablen (wie sie Guo und Bhat (2007) definieren), die von der aggregierten Datenquelle stammen, eingehalten werden. Aus der neuen, angepassten Kreuztabelle wird die multivariate Verteilungsfunktion abgeleitet, indem die Anzahl Agenten jeder Gruppe durch die vorhandene Zahl von Agenten in der Mikroprobe dividiert wird, sodass deren Mengenanteil in der synthetischen Population bestimmt werden kann. Danach werden die Agenten aus der aufgeschlüsselten Stichprobe mit der entsprechenden Wahrscheinlichkeit herauskopiert und in die synthetische Population eingefügt, sodass die aufgeschlüsselte Stichprobe zur Zielmenge expandiert wird.

1.3.1.1 Iterative Proportional Fitting

Das erste, einfachste und meistverwendete Verfahren der synthetischen Rekonstruktion ist das **Iterative Proportional Fitting (IPF),** welches auch als Basis mehrerer Algorithmen dient. Das IPF ist ein Algorithmus, der zur Ermittlung einer multivariaten Verteilung dient und von Deming und Stephan (1940) eingeführt wurde. Die Autoren führten es fälschlicherweise als eine Minimierung der kleinsten Quadrate ein, was Stephan (1942) bemerkte. Bishop *et. al.* (1975) fanden heraus, dass das IPF in Wirklichkeit ein Entropie-maximierendes Verfahren ist. Das IPF ist ein iteratives Verfahren, das benutzt wird, um eine Datentabelle beliebiger Dimension so zu modifizieren, dass die vorgegebenen Randverteilungen eingehalten werden und die erhaltene, angepasste, neue Kreuztabelle derart gestaltet ist, dass sie sich so wenig wie möglich von der ursprünglichen unterscheidet, sodass die Korrelationsstruktur der Daten in optimaler Weise erhalten bleibt.

Der IPF Algorithmus besteht aus einer sich wiederholenden Folge der Anpassung der verschiedenen Dimensionen der Kreuztabelle an deren entsprechende Kontrollvariablen. Dies ergibt sich aus der Tatsache, dass sich jeder Datenpunkt in einem h-dimensionalen Unterraum des $\mathbb{Z}^h$ befindet, dessen Grenzen durch Teilräume von $\mathbb{Z}^{h-1}$ der Dimension h-1 gegeben sind. Diese Teilraume enthalten die Korrelationsstruktur der Attribute, aus denen sie bestehen. Die Anpassung erfolgt dann senkrecht zu jeder Abgrenzung, sodass die Korrelationsstruktur - falls vorhanden - jeder Abgrenzung im ganzen Unterraum des $\mathbb{Z}^h$ erhalten bleibt. Diese Prozedur wird nun kurz für eine Kreuztabelle der Dimension H erläutert, wobei i, j..,p die Anzahl Kategorien, in die jedes Attribut gespalten ist, repräsentieren. Die Datentabelle $\sum_i \sum_{j\ldots} \sum_p a_{i,j,\ldots,p}$ muss an die äusseren Randbedingungen $\sum_i A_i, \sum_j B_j, \ldots, \sum_p H_p$ angepasst werden. Dies wird erreicht, indem die Tabelle mittels mehrerer Iterationen verbessert wird, bis sie sich nur noch in geringer Weise oder bestenfalls gar nicht mehr ändert, d.h. bis eine ausreichende, bzw. totale Konvergenz gegen eine mögliche Lösung erreicht wird. Dabei muss immer die Bedingung $\sum_i A_{i=} \sum_j B_j = \ldots = \sum_p H_p = n$ gelten. Eine Iteration besteht aus einer nacheinander folgenden Adjustierung jeder Dimension an deren entsprechende Kontrollvariablen. Bei einer zweidimensionalen Kreuztabelle $\sum_i \sum_j a_{i,j}$ besteht eine Iteration aus einer Anpassung jeder Zeile, gefolgt von einer Anpassung jeder Spalte an deren Randverteilungen $\sum_i A_i$ und $\sum_j B_j$. Diese beiden Schritte werden so lange wiederholt, bis alle entsprechenden Zellensummen ihre Randbedingungen erfüllen (Abbildung 1.1).

Abbildung 1.1 a und b: Anfangs und Endtabelle vor und nach der Anwendung des IPF

R.B.	B_1	B_2	…	B_j
A_1	a_{11}	a_{12}	…	a_{1j}
A_2	a_{21}	a_{22}	…	a_{2j}
:	:	:	:	:
.	.	.	.	.
A_i	a_{i1}	a_{i2}	…	a_{ij}

(a)

→

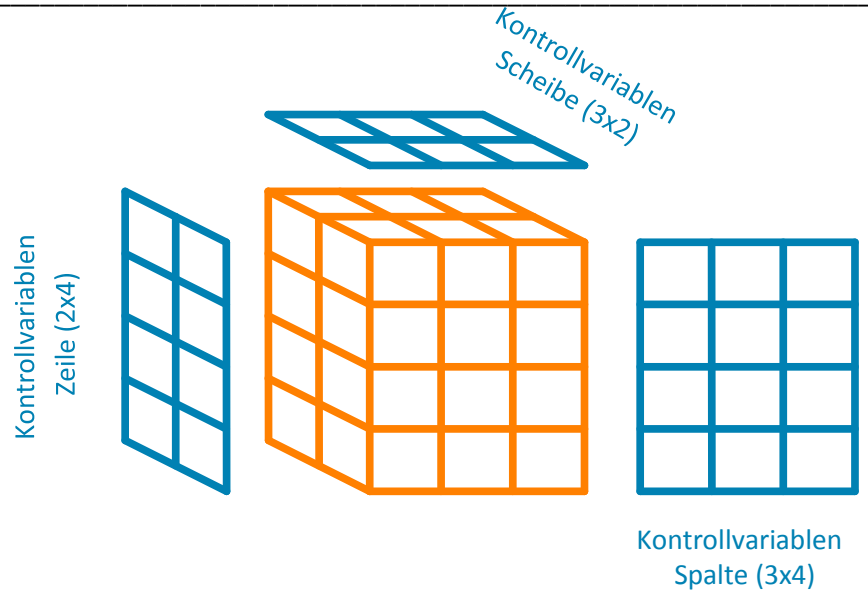
R.B.	B_1	B_2	…	B_j
A_1	a^*_{11}	a^*_{12}	…	a^*_{1j}
A_2	a^*_{21}	a^*_{22}	…	a^*_{2j}
:	:	:	:	:
.	.	.	.	.
A_i	a^*_{i1}	a^*_{i2}	…	a^*_{ij}

(b)

Also bis $\sum_{j=1}^{J} a^*_{ij} \approx A_i$ für alle i und $\sum_{i=1}^{I} a^*_{ij} \approx B_j$ für alle j gilt.

Die Zeilenadjustierung wird durch die Multiplikation jeder Zeile i mit dem Expansionsfaktor $f_i = \frac{A_i}{\sum_{j=1} a_{i,j}}$ erreicht, bzw. die der Spalten j mit $f_j = \frac{B_j}{\sum_{i=1} a_{i,j}}$ (Notation gemäss Müller und Axhausen (2011)). Für drei Dimensionen wird die Kreuztabelle zu einem Würfel, der in Zeilen, Spalten und Scheiben aufgeteilt werden kann (siehe Abbildung 1.2). Dieser Würfel wird dann iterativ an seine Kontrollvariablen angepasst. In diesem Beispiel sind diese paarweise miteinander korreliert. Die Iterationsreihenfolge besteht dann aus drei Schritten: Zuerst werden alle Zeilenzellen an den Wert der Kontrolltabelle (Kontrollvariablenzeile) angepasst. Dies geschieht, indem jede Zelle expandiert wird, sodass die Summe ihrer Werte dem angrenzenden Wert der Kontrollvariablen-Zeilen entspricht. Dasselbe wird dann in einem zweiten Schritt mit den Spaltenzellen gemacht. Als letztes erfolgt eine Anpassung der Scheibenzellen an die Kontrollvariablen-Scheibe.

Abbildung 1.2: Dreidimensionale Kontingenztabelle mit 2D Kontrollvariablen



Beckman *et al.* (1996) benutzten das IPF, um eine synthetische Population aus Haushalten zu erzeugen, unter der Annahme, dass die aufgeschlüsselte Stichprobe und die gesamte Bevölkerung dieselbe Korrelationsstruktur aufweisen. Sie haben den Algorithmus verwendet, um die multivariate Wahrscheinlichkeitsverteilung der Attribute in der Population zu approximieren, indem sie als Anfangskreuztabelle die aufgeschlüsselte Stichprobe (auch „seed data“ genannt) benutzt haben und sie dann mittels IPF an die aggregierten Daten angepasst haben. Aus der resultierenden h-dimensionalen Kreuztabelle, in der die Dimension als Anzahl der Attribute angesehen werden kann, wird die multivariate Wahrscheinlichkeitsmatrix $\sum_{i=1}^{I} \dots \sum_{p=1}^{p} p_{i,j,..,p}$ mit

$p_{i,j,\dots,p} = \frac{Zellenwert\ in\ der\ Kreuztabelle\ eines\ Haushaltstyps}{Anzahl\ der\ Haushalte\ in\ der\ Tabelle}$ abgeleitet. Sie liefert Wahrscheinlichkeit jeder Attributenkombination, mit der dann die finale synthetische Population erstellt werden kann. Durch Multiplikation jedes Wertes $p_{i,j,\dots,p}$ mit der Anzahl zu synthetisierender Agenten wird die erforderliche Anzahl nach einer Aufrundung erhalten. Dies kann auch durch zufälliges Ziehen mit Zurücklegen, mit der gerade berechneten Wahrscheinlichkeit aus der Mikroprobe mittels einer Monte Carlo Simulation erreicht werden (Cho *et al.* (2013)).

Die Populationssynthese mittels IPF kann also in die folgenden 3 Schritte eingeteilt werden (Guo und Bhat (2007)):

1. Festlegung der relevanten Attribute
2. Berechnung der multivariaten Verteilung mittels IPF, basierend auf der aufgeschlüsselten Stichprobe, unter der Bedingung, dass die Kontrollvariablen eingehalten werden.
3. Basierend auf den Proportionen, die in Schritt 2 berechnet werden, werden Haushalte aus der aufgeschlüsselten Stichprobe kopiert und in die synthetische Population eingefügt, bis sie die angestrebte Grösse erreicht hat.

1.3.1.2 Probleme des IPF

Die Populationssynthese mittels IPF nach Beckman *et al.* (1996) ist nicht immer problemlos durchführbar, da bei manchen Datensets keine Konvergenz gewährleistet werden kann. Falls sich im Einflussbereich zweier oder mehrerer Kontrollvariablen nur ein Observationstyp befindet (Abbildung 1.3 a), wird diese Observationsmenge zwischen den Werten dieser schwingen und der IPF kann nicht konvergieren (Müller und Axhausen (2011)).

Abbildung 1.3 (a) Widersprüchliche Kontrollvariablen und (b) Nullzellenproblem

	B_e		
A_g	0.....0	0 : 1 : 0	0.....0

(a)

A_f	0....0	0	0....0

(b)

Abbildung 1.3 a:

Widersprüchliche Kontrollvariablen A_g , B_e

Es ist unmöglich, einen Expansionsfaktor f^* zu finden, so dass

$A_g = f^* * 1 = B_e$ falls $A_g \neq B_e$

(Müller und Axhausen (2011))

Abbildung 1.3 b:

Nullzellenproblem:

Es ist nicht möglich, die Zeile zu expandieren, so dass $f^* * \sum_{j=1}^{J} a_{f,j} = A_f$ gilt, da $f^* = \frac{A_f}{\sum_{j=1}^{J} a_{f,j}} = \frac{A_f}{0}$ nicht existiert (Müller und Axhausen (2011)).

Viel häufiger kommt es vor, dass in der aufgeschlüsselten Stichprobe kein Agent enthalten ist, der einer bestimmten Attributenklasse zugewiesen werden kann, dieser aber in den Kontrollvariablen auftaucht (Abbildung 1.3 b). Dies ist auf die Stichprobeninkonsistenz zurückzuführen. In der Literatur ist dieser Fall als „Nullzellen-Problem“ („zero-cell problem“) bekannt. Bedingt durch eine Division durch null, kann der Algorithmus zu keinem gültigen Resultat kommen. Beckman *et al.* (1996) lösen das Problem mit ihrem „tweaking approach“, bei dem die Nullzellen durch kleine Zahlen (z.B. 0.01) ersetzt werden, womit eine Konvergenz erreicht werden kann. Dies zieht aber als unerwünschten Nebeneffekt eine Verfälschung in der Korrelationsstruktur nach sich, sodass die so künstlich erhaltenen Kombinationen möglicherweise überrepräsentiert werden.

Für die Populationssynthese mittels IPF müssen immer einige Randbedingungen eingehalten werden. Da in den meisten Fällen die Daten, mit denen gearbeitet wird, aus verschiedenen Quellen stammen, kann das Problem der Dateninkonsistenz zwischen verschiedenen Kontrollvariablen entstehen. In diesem Fall weisen korrelierte Kontrollvariablen widersprüchliche Werte auf. Rich und Mulalic (2012) entwickelten ein Verfahren, mit dem es möglich ist, die Kontrollvariablen zu harmonisieren. Anfangend mit der Kontrollvariable, die am relevantesten eingeschätzt wird, werden die anderen mittels einer Korrekturformel so transformiert, dass sie mit der ersten harmonisieren, d.h. konsistent sind.

Die grosse Schwäche des IPF liegt darin, dass es nicht möglich ist, gleichzeitig die Haushalte und die Personen, die in ihnen enthalten sind, an die Randsummen anzupassen. Die Personen werden einfach in das Modell eingefügt, wenn sie einem Haushalt angehören, der in die künstliche Population eingefügt wird, ohne Beachtung der personenbezogenen Kontrollvariablen. Dies führt zu einem Präzisionsverlust, da die haushaltsinterne Korrelationsstruktur nicht kontrolliert wird.

1.3.2 Weiterentwicklungen des IPF mit multiplen Kontrollniveaus

Guo und Bhat (2007), Ye *et al.* (2009), Auld und Mohammadian (2010) und Müller und Axhausen (2011) entwickelten basierend auf dem IPF Algorithmus neue Algorithmen, die es ermöglichen, Populationen zu synthetisieren, in denen gleichzeitig sowohl die Haushaltsverteilungen als auch die Personenverteilungen mit ihren Kontrollvariablen übereinstimmen. Guo und Bhat (2007) und Auld und Mohammadian (2010) erreichen dies mittels einer Anpassung der Auswahlwahrscheinlichkeit der Haushalte, während Ye *et al.* (2009) und Müller und Axhausen (2011) eine neue Strategie anwenden, bei der eine gleichzeitige Anpassung an die Haushalts- und Personen-Randverteilungen möglich ist. Voraussetzung dieser Algorithmen ist das Vorhandensein einer aufgeschlüsselten Stichprobe, bestehend aus Haushalten mit den in ihnen enthaltenen Personen.

1.3.2.1 Guo und Bhat (2007) lösen das Problem durch eine Korrektur des Auswahlkriteriums, nach dem ein Haushalt in die synthetische Population eingefügt wird, sodass auch die marginalen Verteilungen auf Personenniveau übereinstimmen. Als erstes werden die Haushalte h in Gruppen verschiedener Typen aufgeteilt und für diese mittels IPF die Haushaltsgruppen-Expansionsfaktoren f_i^* berechnet. Das Gleiche wird mit den Personen gemacht. Danach werden zwei Kontingenztabellen erstellt, eine für die Haushalte und eine für die Personen. Diese werden dann schrittweise mit Haushalten bzw. Personen gefüllt, bis sie die angestrebte Menge, die durch die Kontrollvariablen vorgegeben wird, erreichen und somit der auf zwei Kontrollebenen angepassten Mikroprobe entsprechen. Sie werden dann, gesteuert durch eine Wahrscheinlichkeitsformel und ein Ausschliesskriterium, mit Haushalten bzw. Personen aus der entsprechenden Mikroprobe gefüllt, bis sie die angestrebte Anzahl Agenten enthalten und somit zur synthetischen Population werden. Die Wahrscheinlichkeit, Haushalt i vom Typ C auszuwählen, ergibt sich aus:

$$P_{i,C} = \frac{w_i}{\sum_{h \in Typ\ C} w_h} * \frac{f_i^* - f_i^{**}}{\sum_j f_j^* - f_j^{**}}$$

Mit

w_i = Haushaltsgewicht des Haushaltes *i* aus der Mikrostichprobe
w_h = Haushaltsgewicht des Haushaltes h aus der Mikrostichprobe
f_i^* = Expansionsfaktor der Haushalte des Typs C
f_i^{**} = Anzahl Haushalte ∈ Typ C in der Tabelle der zu erstellenden Population
f_i^* = Expansionsfaktor für Haushalte ≠ Typ C
f_j^{**} = Wie f_i^{**} aber im Bezug auf den gerade betrachteten Haushaltstyp ≠C

N.B. Je mehr Haushalte eines Typs in der endgültigen Population vorhanden sind, desto mehr sinkt die Wahrscheinlichkeit, dass dieser Typ wiedergewählt wird.

Basierend auf den mit Hilfe des IPF berechneten Wahrscheinlichkeiten $P_{i,C}$, wird ein Haushalt aus der Mikroprobe kopiert und in die Tabelle der zu erstellenden synthetischen Population eingefügt. Dasselbe passiert mit den in ihm enthaltenen Personentypen, die in die Personentabelle eingefügt werden, falls das Auswahlkriterium eingehalten wird. Die Häufigkeit dieses Haushaltstyps und der in ihm enthaltenen Personen muss kleiner sein als ihre vorbestimmten Zielwerte, wobei eine kleine Abweichung zugelassen wird, um eine bessere Konvergenz zu erreichen. Falls ein Haushaltstyp diese Konditionen nicht erfüllt, wird dieser Typ verworfen und nicht mehr berücksichtigt. Es werden solange Haushalte mittels $P_{i,C}$ ausgewählt und eingefügt oder verworfen, bis man die endgültige Grösse der zu erzeugenden synthetischen Population erreicht.

1.3.2.2 Auld und Mohammadian (2010) entwarfen eine neue Wahrscheinlichkeitsformel $P_{i,C}$, mit der die Haushalte in die synthetische Population eingefügt werden. Wie bei Guo und Bhat (2007) werden zu Beginn die multivariaten Verteilungen auf Haushalts- und Personenniveau mittels IPF separat berechnet. Danach wird durch ein iteratives Einfügen der Haushalte die künstliche Population erzeugt. Dies geschieht mit der Auswahlwahrscheinlichkeit $P_{i,C}$:

$$P_{i,c} = \frac{w_i * \prod_{p \in h_i}^{N_{p.Typ}} \frac{f_{i_p}^*}{N_p}}{\sum_{h=1}^{N_C} \left(w_h * \prod_{p \in h_h}^{N_{p.Typ}} \frac{f_{h_p}^*}{N_p} \right)}$$

Mit

w_i, w_h	=	Haushaltsgewichte für Haushalt *i* bzw. *h*
$N_{p.Typ}$	=	Anzahl Personentypen im betrachteten Haushalt
$p \in h_i,\ p \in h_h$	=	Personen die Haushalt *i* oder *h* angehören
$f_{i_p}^*,\ f_{h_p}^*$	=	Expansionsfaktoren der Personen
N_C	=	Anzahl Haushaltstypen
N_p	=	Anzahl Personen in der angepassten Mikroprobe

Diese Formel berücksichtigt beim Synthetisieren der Haushalte explizit die Personenrandverteilungen. Im Gegensatz zur Auswahlformel von Beckman *et. al*. (1996) und Guo und Bhat (2007) haben die neu hinzugefügten Terme die Funktion, dass nun die Wahrscheinlichkeit miteinbezogen werden kann, dass die noch zu erzeugenden Haushalte die fehlenden Personen enthalten.

1.3.2.3 *Ye et al.* (2009) nennen ihren Algorithmus **Iterative Proportional Updating (IPU)**. Der Grundgedanke hinter diesem Verfahren ist, dass die Gewichte $f_{i,h}$ der Haushalte, die die Personen enthalten, iterativ verbessert werden, bis sowohl die Haushaltsverteilungen als auch die Personenverteilungen der Attribute gleichzeitig so gut wie möglich ihren Randverteilungen genügen. Dies geschieht, indem man zuerst alle Haushaltsgewichte $f_{i,h}$=1 setzt und im Folgenden das IPF auf die Haushalte anwendet. Die Personentypen jedes Haushaltes werden mit den gerade berechneten Expansionsfaktoren $f_{i,h}$ multipliziert und danach wird an ihnen wieder das IPF angewendet und so neue $f_{i,h}$ erhalten. Diese Sequenz wird so lange wiederholt, bis eine zufriedenstellende Anpassung erreicht wird. N.B. Es wird nur mit Haushaltsexpansionsfaktoren gearbeitet. Es handelt sich dabei um ein Minimierungsproblem, das von Ye *et al.* (2009) folgendermassen formuliert wird:

Minimiere $\sum_j[(\sum_i N_{a_{i,j}} f_i - A_{i,j})/A_{i,j}]$

Wobei

f_i ≥ 0
a_i = Attribut i
j = Ausprägung eines Attributes
$a_{i,j}$ = Anzahl Agenten mit Attribut a_i in der j Ausprägung
f_i = Expansionsfaktor der i^{ten} Haushaltsgruppe
$A_{i,j}$ = Kontrollvariable des i^{ten} Attributes in der j^{ten} Ausprägung

Das IPU ist ein heuristisches Verfahren, bei dem die erforderliche Konvergenz nicht immer erreicht werden kann. Dies kann vorab jedoch nicht geklärt werden und so muss der Benutzer bestimmen, wann das Iterationsverfahren abgebrochen wird, d.h. wann eine genügende Anpassungsgüte erreicht wird oder das Verfahren zwischen „falschen“ Lösungsscharen gefangen ist.

Das Verfahren wird nun mittels eines kurzen Beispiels verdeutlicht (Abbildung 1.4):

Abbildung 1.4: Beispiel einer Iteration mittels IPU nach Ye et al. (2009)

Haushalts ID	Gewichte	Haushalt Typ 1	Haushalt Typ 2	Personen Typ 1	Personen Typ 2	Gewichte 1	Gewichte 2	Gewichte 3	
1	1	1	0	1	0	11	8.82	8.82	
2	1	1	0	1	2	11	8.82	10.13	
3	1	0	1	1	1	11.33	9.09	10.44	
4	1	0	1	2	1	11.33	9.09	10.44	
5	1	0	1	1	2	11.33	9.09	10.44	
Gewichtete Summe		2	3	6	6				
Zwänge		22	34	54	62				
δ_b		0.91	0.91	0.89	0.9				
Gewichtete Summe 1		22	34	67.32	67.32				
Gewichtete Summe 2		17.64	27.27	54	47.04				
Gewichtete Summe 3		18.95	31.32	60.71	62				
δ		0.14	0.08	0.12	0				

Zunächst wird eine Tabelle erstellt, in die die verschiedenen, anzupassenden Haushalte mit den enthaltenen Personentypen und mit den entsprechenden, einzuhaltenden Randverteilungen eingetragen werden. Das IPU wird dann initialisiert, indem man alle Haushaltsgewichte gleich 1 setzt. Danach beginnt man die erste Iteration, die die Haushaltstypen mittels IPF an die Haushalts-Kontrollvariablen anpasst (Gewichte 1) und die entsprechenden Zeilen damit multipliziert. Im Folgenden werden die Gewichte weiter aktualisiert, indem sie so angepasst werden, dass die Personen ihren Kontrollvariablen genügen. Um dies zu erreichen, werden die gewichteten Personentypen der Reihe nach und in Abhängigkeit des Haushaltes, dem sie angehören, an die entsprechende Kontrollvariable angepasst und dann die entsprechenden Zeilen mit dem erhaltenen Gewicht multipliziert. Somit werden nur Haushaltsgewichte modifiziert, denen die betrachteten Personentypen angehören (Gewichte 2 und 3). Der erste Iterationsschritt ist somit beendet und man beginnt wieder, die Haushalte an ihre Randverteilungen anzupassen, usw. Die Iteration wird dann abgebrochen, wenn die Summe der δ einen gegen null tendierenden Wert erreicht, also wenn nach genügend Schritten:

$$\sum \delta_j \cong 0 \quad \text{mit} \quad \delta_j = \frac{|N a_{i,j} f_i - A_{i,j}|}{A_{i,j}}$$

Ye *et al*. (2009) liefern zudem eine geometrische Interpretation ihrer Methode, bei der erklärt wird, dass der Algorithmus fast wie das Newton Verfahren, beginnend an einem beliebigen Punkt, mit jeder Iteration der gesuchten Lösung näher kommt. Manchmal kann man mit dem IPU nicht die erhofften Gewichte finden, d.h. man erhält keine sinnvolle Lösung, die die Haushalte und die Personen-Attribute gleichzeitig an ihre marginalen Verteilungen anpasst. In diesem Fall schwingt der Algorithmus zwischen zwei Wertescharen.

1.3.2.4 Müller und Axhausen (2011) entwerfen ein dem IPU ähnliches Verfahren, welches unter dem Namen **Hierarchical IPF (HIPF)** bekannt ist. Der HIPF funktioniert analog dem IPU, neu ist, dass auch mit den Personengewichten gearbeitet wird. Dieser Algorithmus basiert auf dem „Principle of Minimum Discrimination Information“ (Kullback and Leibler, 1951; Ireland and Kullback 1968). Dieses Prinzip besagt, dass mit neuen Daten eine neue Distribution gewählt werden sollte, die so schwer wie möglich von der vorherigen zu unterscheiden ist, sodass die neue Distribution den kleinstmöglichen Informationsgewinn produziert. Das HIPF wird, wie der IPU initialisiert, indem die verschiedenen Haushaltsgruppen mittels IPF an ihre Randverteilungen angepasst werden. Die Personen jedes Haushaltes werden mit den gerade berechneten Faktoren gewichtet und der IPF wird dann auf diese angewendet. Daher werden Personenexpansionsfaktoren verwendet, was beim IPU nicht geschieht. Der nächste Schritt besteht aus der Umwandlung der gerade gefundenen Personenexpansionsfaktoren in Haushaltsexpansionsfaktoren, was einem Übergang vom Personenregime ins Haushaltsregime entspricht. Dies wird hier anhand einer Approximation gemacht+ (Für die gesamte Formel siehe Müller und Axhausen (2011)):

$$f_{i,h} := \frac{1}{p_h} \sum_{p \in P_h} f_{i,p}$$

Mit

$f_{i,h}$ = Haushaltsexpansionsfaktoren
$f_{i,p}$ = Personenexpansionsfaktoren
p_h = Anzahl Personen im Haushalt P_h

Diese Iteration wird so lange wiederholt, bis die gewünschte Genauigkeit erreicht wird.

Beispiel einer Iteration mittels HIPF:

Die folgende Starttabelle wird durch HIPF an ihre Kontrollvariablen angepasst.

Abbildung 1.5 a, b und c: Beispiel einer Iteration mittels HIPF gemäss Müller und Axhausen (2011)

Haushalt ID	Typ 1	Typ 2	p_h	Typ1	Typ 2
1	1	0	3	2	1
2	1	0	1	0	1
3	0	1	2	1	1
4	0	1	3	3	0
Kontrollvariablen:	12	18		38	27

(a)

1) Die Haushalte Typ 1 und 2 werden mittels IPF an ihre Kontrollvariablen angepasst:
$f_{1,h} = 12 \div (1 + 1) = 6$
$f_{2,h} = 18 \div (1 + 1) = 9$
und die entsprechenden Zeilen damit multipliziert.

Haushalt ID	Typ 1	Typ 2	Anzahl Personen	Typ1	Typ 2
1	6	0	3	12	6
2	6	0	1	0	6
3	0	9	2	9	9
4	0	9	3	27	0
Kontrollvariablen:	12	18		38	27

(b)

2) Die mit den vorab bestimmten Expansionsfaktoren $f_{1,h}$, $f_{2,h}$ gewichteten Personen werden nun mittels IPF an ihre Kontrollvariablen angepasst:
$f_{1,p} = 38 \div (12 + 9 + 27) = 0.79$
$f_{2,p} = 27 \div (6 + 6 + 9) = 1.29$

und die somit erhaltenen Gewichte $f_{1,p}$ und $f_{2,p}$ für jeden Haushaltstyp in dessen Dimension rücktransformiert (Approximation):

$f_{1,h} = \frac{1}{3}(.79 * 2 + 1.29 * 1) = 0.96$, $f_{2,h} = \frac{1}{1}(.79 * 0 + 1.29 * 1) = 1.29$,
$f_{3,h} = \frac{1}{3}(.79 * 1 + 1.29 * 1) = 1.04$, $f_{4,h} = \frac{1}{3}(.79 * 3 + 1.29 * 0) = 0.79$

und die entsprechenden Zeilen damit multipliziert. Man erhält somit die verbesserte Tabelle:

Haushalt ID	Typ 1	Typ 2	Anzahl Personen	Typ1	Typ 2
1	5.7	0	3	2	1
2	7.68	0	1	0	1
3	0	9.36	2	1	1
4	0	7.11	3	3	0
Kontrollvariablen:	12	18		38	27

(c)

Diese Tabelle dient als Starttabelle für die nächste Iteration. Es wird solange weitergemacht, bis die Tabelle allen Kontrollvariablen genügt.

1.3.3 Probleme des IPU und des HIPF

Die beiden Synthetisierungs-Methoden bergen Probleme, die im Folgenden dargestellt werden. Diese Probleme sind in der Literatur oft behandelt worden wie z.B. von Cho *et al*. (2013), Müller und Axhausen (2011), Ye *et al.* (2009). Das Nullzellen-Problem und das Null-Marginal-Problem führen dazu, dass das HIPF und der IPU zu keinem Ergebnis kommen. Das **Nullzellen Problem** wurde schon erläutert (siehe oben), es wird jedoch darauf hingewiesen, dass Ye *et al.* (2009) diese Angelegenheit mit einer Abschätzung mittels der aggregierten Daten zu lösen versuchen. Der Wert, durch den die Nullzelle ersetzt wird, wird bestimmt, indem man annimmt, dass die Häufigkeit der Gruppe gleich ist, wie in der aggregierten Datentabelle. Auch diese Methode kann zu einer Überrepräsentierung der Gruppe führen.

Das **Null-Marginal Problem** tritt hingegen auf, wenn der Randwert einer Attributen-Dimension null ist. Die IPF Prozedur weist so jedem Haushalts-/Personentyp dieser Dimensionskategorie ein Nullgewicht zu und hat zur Folge, dass das IPU und das HIPF diese nicht in die synthetische Population miteinbeziehen. Ye *et al.* (2009) umgehen diese Problematik, indem sie jeder Nullkontrollvariable eine kleine Zahl, z.B. 0.01 zuteilen. Sie kommen zu dem Schluss, dass deren Einfluss auf die Lösung nach ein paar Iterationen vernachlässigbar ist und die Verfahren somit zu einer Lösung konvergieren können.

1.3.4 Combinatorial Optimisation (CO)

Diese, im Gegensatz zu den synthetischen Rekonstruktionen selten verwendete Technik zur Populationssynthese, wurde zum ersten Mal von Williamson *et al.* (1998) vorgestellt. Auch mit ihr ist es möglich, den Datensatz an multiple Kontrollniveaus anzupassen. Man hat es auch hier mit einem iterativen Prozess zu tun, der folgendermassen abläuft: Zuerst wird ein Agentensatz zusammengestellt, - normalerweise aus Haushalten mit den enthalten Personen - indem zufällig aus der aufgeschlüsselten Stichprobe Agenten herauskopiert werden, bis sie der Menge entsprechen, die in den aggregierten Daten erwartet wird. Dieser Satz wird dann Schritt für Schritt verbessert, bis eine Kombination gefunden wird, die ausreichend gut mit den Randverteilungen kompatibel ist. Diese verbesserte Kombination wird erreicht, indem man einen Agenten des Satzes zufällig durch einen anderen aus der Stichprobe ersetzt. Falls der Anpassungsgrad an die Randverteilungen steigt, wird der Austausch gespeichert, falls nicht, wird er gelöscht und vom vorherigem Standpunkt weitergemacht. Dieser Prozess wird dann so lange wiederholt, bis man die gewünschte Anpassungsgüte erreicht. Diese wird nach Voas und Williamson (2001, p. 187) mittels dem „overall relative sum of squared Z scores“ RSSZ-Wert bestimmt. Dabei bedeutet eine Verringerung dieses Wertes eine bessere Anpassung und somit einen Austausch. Der RSSZ wird folgendermassen definiert:

$$RSSZ = \sum_i SSZ_i,$$

Dieser Wert ist ein Mass für die aufsummierte Abweichung der Anzahl jeder Attributenkategorie von der entsprechenden Kontrollvariable.

$$SSZ_i = \sum_i F_{i,j}(a_{i,j} - A_{i,j})^2$$

$$F_{j,i} = \begin{cases} \left(C_i a_{i,j}\left(1 - \frac{a_{i,j}}{N_i}\right)\right)^{-1} & \text{, wenn } a_{i,j} \neq 0 \\ \frac{1}{C_i} & \text{, wenn } a_{i,j} = 0 \end{cases}$$

Wobei

$a_{i,j}$ = Anzahl des im Satzes enthaltenen Agenten mit Attribut a_i in der Ausprägung j
$A_{i,j}$ = Kontrollvariable des Attributes $a_{i,j}$
C_i = 5% $\varkappa^2$ kritischer Wert des Attributes a_i
N_i = Anzahl Agenten

Der RSSZ Wert sollte gegen den kleinstmöglichen Wert tendieren, was bedeuten würde, dass die so erhaltene Kombination bestens mit den Kontrollvariablen übereinstimmt. In der Praxis lässt man nach Ryan *et al. (*2009) den Algorithmus solange arbeiten, bis alle $SSZ_k \leq \mathcal{E}$ mit z.B. $\mathcal{E}$=0.1, sodass der RSSZ einen kleinen Wert hat. Wenn die Endkombination der Agenten gefunden ist, wird wie beim IPF vorgegangen und sie somit zur gesuchten Population expandiert.

1.3.5 Vergleich von synthetischer Rekonstruktion und kombinatorischer Optimisation

Ryan *et al. (*2009) haben den Einfluss der Grösse der aufgeschlüsselten Stichprobe auf synthetische Populationen, die mittels IPF und mittels CO erzeugt wurden, analysiert und verglichen. Ihre Studie basiert auf einem kompletten Datensatz mit allen relevanten Informationen einer Firmen-Population. Sie entnehmen der Firmen-Population in ihrem Experiment verschieden grosse Stichproben und expandieren diese nach einer Anpassung mittels IPF und CO zur synthetischen Population. Auch der Einfluss der Attributendimension wird analysiert. Das Ergebnis der Studie kann in drei Aussagen zusammengefasst werden.

1. Bei beiden Methoden ist mit steigender Stichprobengrösse eine Qualitätssteigerung der synthetisierten Population zu erkennen, welche aber bei Intervallvergrösserung der Stichprobe nicht einheitlich ausfällt.
2. Mit steigender Attributenanzahl steigt bei beiden Synthetisierungsmethoden die Qualität der synthetisierten Population, die mittels der Freeman-Tukey Statistik gemessen wird.
3. Mit der CO Methode werden für kleine Stichproben genauere Populationen geringerer Varianz synthetisiert.

Es muss noch erwähnt werden, dass mit kleinen Populationen gearbeitet wurde, bei denen die Konvergenzzeit keine Rolle spielte. Da in der Praxis mit enormen Datenmengen gearbeitet wird, wird die CO Methode wegen des grossen Zeitbedarfs für das Erreichen einer ausreichenden Konvergenz meistens verworfen.

1.4 Abschätzung fehlender Kontrollvariablen

Da alle vorab vorgestellten Algorithmen für jede Attributenklasse die entsprechenden Kontrollvariablen benötigen, diese aber nicht immer vorhanden sind, wird von Wongchavalidkul *et al.* (2009) ein Verfahren entworfen, um diese zuverlässig abzuschätzen. Diese Technik findet die fehlenden Randverteilungen, indem sie die Quadrate der Fehler zwischen den bedingt geschätzten Wahrscheinlichkeiten und den bedingten Wahrscheinlichkeiten der Zielvariablen minimiert (Methode der kleinsten Quadrate), die aus der aufgeschlüsselten Stichprobe und den vorhandenen Kontrollvariablen resultieren.

Sei I×J×K eine 3-dimensionale Kontingenztabelle, X, Y, Z diskrete Zufallsvariablen, die folgende Werte annehmen können $\{x_1, x_2, \dots, x_i\}$, $\{y_1, y_2, \dots, y_j\}$, $\{z_1, z_2, \dots, z_k\}$, und seien $n^*_{ij\dots q}$ die observierten Zellenwerte aus den Beobachtungen in der Kontingenztabelle.

Die totale Population entspricht dann $N = \sum_{k=1}^{K} \sum_{j=1}^{J} \sum_{i=1}^{I} n_{ijk}$ und die totale Population einer spezifischen Kategorie $n_{i..} = \sum_{j=1}^{J} \sum_{k=1}^{K} n_{ijk}$ oder $n_{.j.} = \sum_{i=1}^{I} \sum_{k=1}^{K} n_{ijk}$ oder $n_{..k} = \sum_{i=1}^{I} \sum_{j=1}^{J} n_{ijk}$

Das Problem kann also folgenderweise umformuliert und mittels der Methode der kleinsten Quadrate gelöst werden. Dabei sind immer die Kontrollvariablen der Ausprägungen von zwei von drei Attributen $n_{i..}, n_{.j.}, n_{..k}$ gegeben:

Minimiere $\{(\sum a^*_{ijk} - \sum a_{ijk})^2 + (\sum b^*_{ijk} - \sum b_{ijk})^2 + (\sum c^*_{ijk} - \sum c_{ijk})^2\}$

mit

$$a_{ijk} = P\big(X = x_i \mid Y = y_j, Z = z_k\big) = \frac{n_{ijk}}{\sum_{k=1}^{K} \sum_{j=1}^{J} n_{ijk}},$$

$$b_{ijk} = P\big(Y = y_j \mid X = x_i, Z = z_k\big) = \frac{n_{ijk}}{\sum_{k=1}^{K} \sum_{i=1}^{I} n_{ijk}},$$

$$c_{ijk} = P\big(Z = z_k \mid X = x_i, Y = y_j\big) = \frac{n_{ijk}}{\sum_{j=1}^{J} \sum_{i=1}^{I} n_{ijk}},$$

die bedingte, geschätzte Wahrscheinlichkeiten sind und $a^*_{ijk}, b^*_{ijk}, c^*_{ijk}$ die bedingten Wahrscheinlichkeiten aus den Daten der Stichproben.

1.5 Neuere Techniken

Diese Methoden haben den Vorteil, dass sie weniger von der Stichprobenqualität abhängen und in der Lage sind, Agenten zu erzeugen, die nicht in der aufgeschlüsselten Stichprobe enthalten sind, aber trotzdem mit hoher Wahrscheinlichkeit existieren. Auch diese Methoden erstellen eine multivariate Wahrscheinlichkeitsverteilung mit der die Agenten dann direkt synthetisiert werden.

1.5.1 Die auf Kopulas basierende Methode

Kao *et. al.* (2013) versuchen, die multivariate Verteilung der Attribute mittels Copulas aus den Stichproben abzuleiten. Copulas sind Funktionen, die benutzt werden, um den funktionalen Zusammenhang zwischen den Randverteilungsfunktionen verschiedener Zufallsvariablen und ihrer gemeinsamen Wahrscheinlichkeitsverteilung zu finden. Nach Sklar ist es möglich, multivariate Verteilungsfunktionen aus der Kombination von eindimensionalen Funktionen zu generieren. Er hat bewiesen, dass es genau eine d-Copula $C_{U_1,U_2,\dots,U_d}$, gibt, so dass $H_{X_1,X_{2,\dots},X_d}(x_1,x_2,\dots,x_d) = C_{U_1,U_2,\dots,U_d}(u_1,u_{2,\dots},u_d)$ gilt. $u_j = F_{X_j}(x_j)$, j=1,...,d repräsentieren die marginalen kumulativen Distributionsfunktionen der Variablen $\{X_1, X_{2,\dots}X_d\}$ und können als eine Transformation von $[-\infty,\infty] \rightarrow [0,1]$ angesehen werden. Somit erhält man eine komplette, mathematische Charakterisierung der gesamten Abhängigkeitsstruktur. Kao *et. al.* (2013) erfassen die Korrelationsstruktur der Attribute aus der Mikroprobe mittels einer Gausscopula. Diese erlaubt multidimensionale Variablen zu behandeln. Dabei ist zu beachten, dass die kategorischen Attribute erst nach der Berechnung mittels der Gausskopula miteinbezogen und mit ihr kombiniert werden. Über die Copulafunktion wird die Abhängigkeit einer Attributenausprägung von den anderen indem quantifiziert und somit jeder Attributenkombination eine Wahrscheinlichkeit zugewiesen. Die Attributenkombinationen werden mittels Variation der Randverteilungen erhalten. Sie werden aus den aggregierten Stichproben ermittelt. Dann werden virtuelle Haushalte anhand ihrer Wahrscheinlichkeit synthetisiert und in eine Tabelle eingefügt, deren Struktur identisch derer der Mikroprobe ist. Dies passiert aber nur solange, bis die maximale Anzahl einer Attributenausprägung erreicht wird, die von der entsprechenden Kontrollvariable vorgegeben wird. Danach werden alle virtuellen Haushalte verworfen, die zu einem Überschuss dieser Attributenausprägung führen und nur noch Haushalte verwendet, die benötigt werden, um die noch nicht ausgeschöpften Attributenkategorien zu füllen und die zu keinem Überschuss führen. Mittels dieser Prozedur wird eine „komplettere“ Mikroprobe erhalten, welche im Vergleich zur aufgeschlüsselten Stichprobe weniger Nullzellen enthält. Diese Menge von Haushalten wird dann zur synthetischen Population expandiert.

1.5.2 Die Markov Chain Monte Carlo Methode (MCMC)

Farooq *et al.* (2013) schätzen die multivariate Wahrscheinlichkeitsverteilung der Attribute anhand der zur Verfügung stehenden, bedingten Wahrscheinlichkeiten. Man nutzt aus, dass diese als partieller Einblick in die gesamte Verteilung angesehen werden können. Dies wird durch den von Geman und Geman (1984) entwickelten Gibbs-Sampling Algorithmus möglich. Dieser MCMC-Algorithmus erfasst die Abhängigkeit zwischen den Variablen aus den bedingten Wahrscheinlichkeiten, die als Stichproben der realen Verteilung fungieren, und verknüpft sie in einer Form, die es ermöglicht, die reale Verteilung zu simulieren. Diese Methode besitzt den Vorteil, dass nicht zwingend eine Mikroprobe erforderlich ist und dass verschiedene Datenquellen ohne Probleme eingebunden werden können. Um das Gibbs-Sampling anwenden zu können, muss ein Datensatz erstellt werden, der die bedingten Wahrscheinlichkeiten jedes Attributes in Abhängigkeit zu den Ausprägungen aller anderen enthält. Da dies nicht immer möglich ist, werden diese künstlich erzeugt. Dies geschieht unter der Annahme, dass Attributenausprägungen nur von denen abhängen, deren Abhängigkeit durch die zu Verfügung stehenden Datenquellen quantifiziert wird, oder daraus abgeleitet werden kann. Es können auch komplette, bedingte Wahrscheinlichkeiten aus logischen Betrachtungen erstellt werden. Z.B. wird die bedingte Wahrscheinlichkeit, dass eine Person, die minderjährig ist, 1000000 CHF im Jahr verdient, gleich null gesetzt usw.

2 Populationssynthese mittels CART Regressionsbäumen

2.1. Einleitung

Dieser zweite Teil der Arbeit gibt einen kurzen Einblick in die Methodologie des auf CART Regressionsbäumen basierenden Algorithmus und die Theorie, auf der er aufbaut. Im Hauptteil werden dann die Ergebnisse verschiedener Parameterkombinationen, die den Algorithmus steuern, analysiert, mit dem Ziel, die beste zu finden. Dies wird im Rahmen eines Experiments geschehen.

2.2 Der Algorithmus

Die Funktion dieses parameter-abhängigen Algorithmus ist die Konstruktion einer statistisch konsistenten, synthetischen Population, basierend auf CART Regressionsbäumen (Classification and Regression Trees). Es wird versucht, mittels dieser Entscheidungsbäume die n-dimensionale, multivariate Verteilungsfunktion $P(\boldsymbol{X})$ der Attribute $\boldsymbol{X} = (X_1, X_2, \dots, X_n)$ der gesamten Population so gut wie möglich abzuschätzen, ausgehend von einer bedeutend kleineren Teilmenge $\boldsymbol{X}^* = (X_1^*, X_2^*, \dots, X_n^*) \subset \boldsymbol{X}$. Man nutzt dabei die Eigenschaft aus, dass die Verteilungsfunktion $P(\boldsymbol{X})$ in die Form $P(\boldsymbol{X}) = P(X_1) * P(X_2 \mid X_1) * \dots * P(X_n \mid X_1, X_2, \dots, X_{n-1})$ [1] umgeschrieben werden kann. Die Populationssynthese mittels Regressionsbäumen kann in zwei Phasen unterteilt werden. In der ersten wird die multivariate Wahrscheinlichkeitsverteilung der Attribute aus der Stichprobe geschätzt und in der zweiten wird mit ihr die synthetische Population erzeugt. Die Verteilung wird in n Schritten berechnet, einem für jeden Term von Gleichung [1]. Die Reihenfolge der behandelten Attribute und der damit gebildeten Bäume ist fix und bei der Berechnung von [1] wird immer anhand desselben Musters vorgegangen. Erstens wird die Auftretenswahrscheinlichkeit $P(X)$ jeder Ausprägung von X_1 ermittelt. Danach wird der zweite Term $P(X_2 \mid X_1)$ der Gleichung ermittelt, also die Wahrscheinlichkeiten der Ausprägungen von X_2 bei gegebener Ausprägung von X_1. Es wird so weitergemacht, bis die Wahrscheinlichkeit jeder Attributenkombination berechnet worden ist, indem die Wahrscheinlichkeit $P(X_i \mid X_1, X_2, \dots, X_{i-1})$ aller Ausprägungen des gerade behandelten Attributes in Abhängigkeit aller Kombinationen der vorherigen ermittelt wird. Diese bedingten Wahrscheinlichkeiten erhält man durch die Bildung von Regressionsbäumen, mit denen auch bestimmt wird, von welchen vorherigen Attributen das gerade behandelte abhängt. Nachdem Gleichung [1] für jede mögliche Kombination ausgewertet worden ist, weist der Algorithmus jeder Attributenkombination ein Gewicht zu, das deren Auftretenswahrscheinlichkeit wiederspiegelt. Die synthetischen Agenten werden dann jeder Attributenkombination zugewiesen, in der Anzahl, die sich aus der Multiplikation jeder Attributenkombination mit der Anzahl der insgesamt zu erzeugenden Agenten ergibt (Phase 2).

Diese Vorgehensweise besitzt im Vergleich zu den „Resampling“ Modellen die Eigenschaft, dass neue Attributenkombinationen erzeugt werden, die nicht in der Stichprobe vorhanden, aber wahrscheinlich sind (Müller und Flötteröd (2013)).

2.3 CART Regressionsbäume

CART Regressionsbäume sind das Ergebnis der Arbeit von Breiman *et al.* (1984). Diese Bäume werden benutzt, um eine d-dimensionale Observationsmenge $\mathbb{O}$, bestehend aus gleichlangen Attributenkombinationen aus kategorischen Attributen, mittels Entscheidungsbäumen hierarchisch zu ordnen. Da alle Attribute aus einer Menge von Klassifizierungsmerkmalen bestehen, denen eine Nummer $n \in \mathbb{R}$ zugewiesen wird, kann jede Observation als ein numerischer Vektor aufgefasst werden. Jede Observation $O = (X_1 = x_1;\ X_2 = x_2, \dots, X_d = x_d)$ kann somit mittels eines Punktes im Raum $\mathbb{R}^d$ beschrieben werden. Dazu müssen die verschiedenen Attribute entsprechend vorbereitet werden. Jeder Ausprägung eines nominalen Attributes wird eine Zahl zugewiesen, die einfach als ihre Beschreibung dient. Ordinale Attribute hingegen müssen zuerst in Intervalle aufgeteilt werden, denen dann in Reihenfolge eine Zahl zugewiesen wird, die das Intervall eindeutig definiert. Die Observationsmenge wird dann mittels einer Regressionsbaumbildung, weiterhin binär, in Regionen höchster Homogenität gespalten.

$\mathbb{O} = \mathbb{R}^d \rightarrow \mathbb{O}_{links} \cup \mathbb{O}_{rechts}$

wobei:

$\mathbb{O}_{links} = \mathbb{R} \times \mathbb{R} \times \dots \times (-\infty, s] \times \mathbb{R} \times \dots \times \mathbb{R}$ und $\mathbb{O}_{rechts} = \mathbb{R} \times \mathbb{R} \times \dots \times (s, \infty) \times \mathbb{R} \times \dots \times \mathbb{R}$

Diese progressive Spaltung wird mittels einer bestimmten Bedingung vollzogen, die entweder eingehalten wird oder nicht. Alle Observationen, die der Bedingung genügen, werden zu einer neuen Untermenge $\mathbb{O}_{links}$ zusammengefasst, während mit denen, die ihr nicht entsprechen, die zweite Teilmenge $\mathbb{O}_{rechts}$ gebildet wird. Dasselbe wird dann so lange mit allen erzeugten Untermengen gemacht, bis diese nur noch aus gleichen Observationen bestehen. In diesem Fall wird der maximale Baum erreicht. N.B. Die Endteilmengen können nach verschiedenen Teilungsschritten erhalten werden. Jede Observation aus der Ausgangsmenge wandert somit immer tiefer im Baum, bis sie ihre Endteilmenge erreicht. Als nächstes muss nun die Bedingung festgelegt werden, die es erlaubt, alle Punkte im betrachteten Raum oder Teilraum des $\mathbb{R}^d$ in zwei Regionen grösster Homogenität zu spalten. Diese erhält man, wenn die negative Log-likelihood maximal reduziert wird. In diesem Falle ist die Summe der Varianz aller Punkte in den zwei neuen Teilräumen bezüglich der zu findenden Koordinate die kleinstmögliche. Somit kann der Regressionsbaum über die progressive Spaltung jedes Raumes mittels der Lösung von [2] erhalten werden.

Finde j und s, so dass $\sum_{X_j \leq s}(\bar{X}_j - X_j)^2 + \sum_{X_j > s}(\bar{X}_j - X_j)^2$ minimal wird. [2]

Mit

X_j = zu bestimmende Koordinatenachse
$\bar{X}_j$ = Mittelwert aller Punkte bezüglich Koordinatenachse j
s = zu bestimmender numerischer Wert

Mit anderen Worten besteht das Problem in der Identifizierung des Attributes X_j, durch dessen Spaltung die Menge am homogensten geteilt werden kann.

Diese CART Regressionsbäume haben die positive Eigenschaft, dass die Wichtigkeit jeder Attributenspaltung berechnet werden kann, sodass eine hierarchische Ordnung entsteht, die in Form eines Baumes visualisiert wird. Die Wichtigkeit einer Spaltung bemisst sich darin, dass über sie der höchstmögliche Klassifikationsgewinn erhalten wird, weil die zwei grösstmöglichen Teilmengen entstehen. Die Wichtigkeit jeder Attributendivision spiegelt sich in ihrer Platzierung im Baum wieder: je höher desto wichtiger.

2.3.1 Tree Pruning

Falls Bedingung [2] uneingeschränkt angewendet wird, erhält man den maximalen Baum $T_{M_{max}}$. Dieser Baum ist aber überangepasst („overfittet“). Mit diesem negativen Begriff wird zum Ausdruck gebracht, dass ein Modell in übertriebener Weise an die Daten angepasst ist, mit denen es erstellt wurde, und daher für die Anwendung nur bedingt geeignet ist. Eine Möglichkeit, dieses Problem zu lösen, besteht darin, Teile des Baumes abzutrennen, also das „pruning“. Dabei werden die Endknoten Schritt für Schritt zusammengefügt, was die kleinste Vergrösserung der negativen Log-likelihood zur Folge hat. Dies wird so lange wiederholt, bis der Regressionsbaum die optimale Grösse aufweist. Diese wird anhand der Minimierung von Formel [3] bestimmt.

$$R_{\alpha}(T) = R(T) + \alpha * \text{size(T)},\ \alpha \geq 0 \qquad [3]$$

Wobei

$R(T)$ = Varianz
$size(T)$ = 1 + Anzahl „Teilungen“
α = zu bestimmender Parameter

Der Parameter $\propto$ wird mittels „cross Validierung“ bestimmt.

2.4.1 Das Experiment

Bevor die Leistungsfähigkeit des Algorithmus getestet werden kann, muss die bestmögliche Parameterkombination gefunden werden, also diejenige, die den Algorithmus am effizientesten steuert. Um dies zu erreichen, muss zunächst der Einfluss dieser Parameter studiert und verstanden werden. Die Auswirkung aller Kombinationen dieser wird zunächst analysiert, indem ihr Einfluss auf die Qualität der damit erzeugten Population gemessen wird. Damit dies möglich wird, müsste man die komplette Population, aus der die Stichprobe stammt, zur Verfügung haben. Nur so hätte man ein Vergleichsmass zur Hand. Dieses Szenario wird erreicht, indem man vorgibt, dass die zur Verfügung stehende 5% Mikroprobe die komplette Population darstellt. Diese besteht aus 349792 Observationen in Bezug auf 13 Attribute und die synthetische Rekonstruktion der Population erfolgt anhand Stichproben der Mikroprobe. Die Daten, mit denen gearbeitet wird, sind die des Schweizer Public Use Microsample (PUMS) des Jahres 2000. Das Experiment, mit dessen Hilfe das Verständnis der Parameter möglich wird, läuft folgendermassen ab: Für jede mögliche Parameterkombination werden 100 Simulationen

mit 100 verschiedenen, zufällig erstellten Stichproben durchgeführt und in verschiedenen Szenarien wiederholt. Die Durchschnittswerte und die Standardabweichung der Indikatoren, mit denen die Qualität der synthetisierten Populationen angegeben wird, werden dann betrachtet und mit denen anderer Parameterkombinationen verglichen, mit dem Ziel, Trends zu entdecken und die beste Kombination für jedes Szenario zu finden.

2.4.2 Die Szenarien

Die Parameterkombinationen, die den Algorithmus steuern, können in verschiedenen Szenarien getestet werden. Sie werden anhand der Variation der Stichprobengrösse und der Anzahl berücksichtigter Attribute unterschieden.

- Die Stichprobengrösse kann mittels r ∈ [0.01; 0.02; 0.05; 0.1; 0.2] gewählt werden. Dabei steht r für die Stichprobengrösse, bezogen auf die vollständige Population.
- Die Anzahl berücksichtigter Attribute kann mittels ncols ∈ [3; 13] variiert werden, wobei ncols=13 aufgrund der langen Rechenzeit nicht betrachtet wird.

2.4.3 Die Parameter

Der Algorithmus wird mittels folgender Parameter gesteuert:

- Der **Weight**parameter ∈ [TRUE; FALSE]:
  Es handelt sich dabei um ein Implementationsdetail, mit dem man wählen kann, ob für die Modellschätzung die tabulierten Daten mit Gewichten verwendet werden (TRUE), oder die Daten vorher expandiert werden und mit Einheitsgewichten gearbeitet wird (FALSE).
- Der Komplexitätsparameter **CP** ∈ [-1; 0; 0.005; 0.01; 0.02]:
  Dieser ermöglicht es, Rechenzeit zu sparen, indem Splits, die das Endergebnis nur sehr geringfügig beeinflussen, nicht durchgeführt werden. Der Komplexitätsparameter bestimmt somit die Grösse und die Präzision der erstellten Regressionsbäume, indem er einen Richtwert vorgibt, der quantifiziert, in welchem Mass sich eine Teilung mindestens lohnen muss, damit sie durchgeführt wird. CP=-1 ist dabei ein Spezialfall. Mit ihm werden immer die grösstmöglichen Bäume erzeugt. Grossen cp Werten hingegen folgen kleinere Bäume, da weniger Teilungen durchgeführt werden.
- Der Filterparameter **MM** ∈ [0; 1; 2; 3]:
  Mit diesem Parameter wird bestimmt, welche vom Programm neu erzeugten Kombinationen aussortiert werden. Dies wird erreicht, indem alle Attributen-kombinationen, die ein nicht in der Stichprobe enthaltenes Tripel (MM=3) oder Paar (MM=2) von Attributenkategorien-Kombinationen enthalten, aussortiert werden. MM=1 sortiert alle Kombinationen aus, die eine Kategorienrealisierung beinhalten, die es nicht in der Stichprobe gibt. Bei MM=0 erfolgt keine Aussortierung. Beispiel: Ist in der Mikroprobe keine Person enthalten, die minderjährig und berufstätig ist und ein jährliches Einkommen zwischen 1.000.000 und 2.000.000 CHF besitzt, werden bei MM=3 alle erzeugten Kombinationen, die diese drei Ausprägungen gleichzeitig enthalten, gestrichen. Dies geschieht analog bei MM=2, MM=1 und MM=0, mit dem Unterschied, dass die Ausprägungen nur paarweise bzw. einzeln oder gar nicht betrachtet werden.

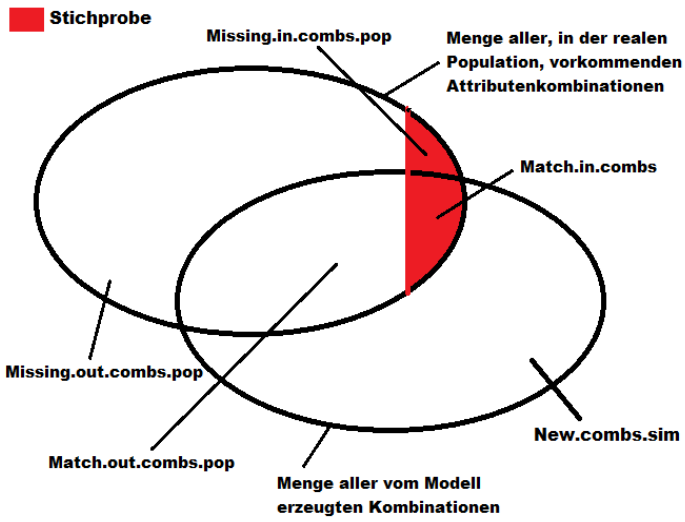
2.4.4 Die Stichproben

Aus der Stichprobe der realen Schweizer Population werden 100 zufällige Stichproben erstellt. Diese werden mit einer Permutationsmatrix gezogen. Da die gesamte Population in einer Tabelle aufgelistet ist, in der jede Zeile einer Person der Stichprobe samt ihrer Attributenkombination entspricht, werden mittels dieser Matrix die Zeilen ausgewählt, die in die Stichprobe miteinbezogen werden. Die erhaltenen Stichproben können in der geforderten Grösse anhand des Parameters r gewählt werden, mit dem die Anzahl gezogener Zeilen festgelegt wird.

2.4.5 Die Indikatoren

Um die Qualität einer synthetisierten Population messen zu können, muss diese zunächst definiert werden. Eine qualitativ hochwertige, synthetische Population sollte weitest möglich der realen gleichen. Um den Ähnlichkeitsgrad zu bestimmen, sind Mengen-Diagramme das beste Werkzeug, da ihre Überlappungsregionen die Teilmengen enthalten, die beiden Populationen angehören. Es werden vier Diagramme erstellt, in denen die synthetische Population mit der realen auf verschiedenen Ebenen verglichen wird. Jede Observation in der realen Population entspricht einer durch ihre Eigenschaften beschriebenen Person, während eine Observation der simulierten Population einem synthetischen Agenten entspricht. Der erste Vergleich findet auf der Kombinationsebene statt (Abbildung 2.1).

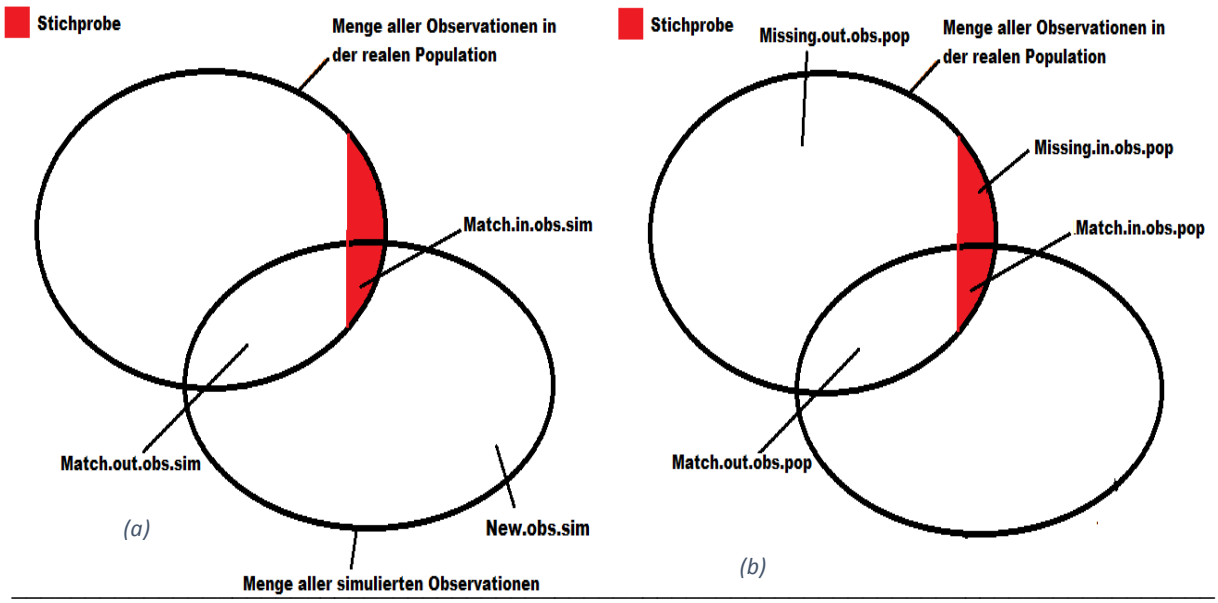
Abbildung 2.1: Kombinationsindikatoren



Es ist offensichtlich, dass eine gute, synthetische Population zwei Eigenschaften besitzen sollte. Erstens sollte die Kombinationsüberlappungs-Region mit der realen Population grösstmöglich sein (*match.in.combs* + *match.out.combs*). „*In*“ und „*out*“ bedeuten: in der Stichprobe vorhanden oder nicht. Das heisst, dass die simulierte Kombinationsmenge, bestehend aus allen vom Modell erzeugten Kombinationen, der der realen Population weitmöglich entspricht. Da dies nie der Fall sein wird, müssen zweitens die Mengen der neusimulierten Kombinationen, die nicht in der realen Population auftreten (*new.combs.sim*) und die der nicht erfassten (*missing.in.combs* und *missing.out.combs*) so klein wie möglich sein. Die zwei genannten Charakteristika sind für eine Analyse der synthetischen Population aber nur bedingt aussagekräftig, da noch in Betracht gezogen werden muss, dass die Kombinationen mit einem

Gewicht ausgestattet sind. Der Algorithmus weist jeder in der Stichprobe enthaltenen Kombination und den neu erzeugten, bedingt durch Fehler des Algorithmus (nicht als negativ zu bewerten), ein Gewicht zu, welches bei der Erzeugung der synthetischen Agenten sichtbar wird. Deshalb führt eine grosse Anzahl neuer Kombinationen nicht zwangsläufig zu einer hohen Anzahl neuer Agenten, wenn diese Kombinationen nur ein sehr kleines Gewicht besitzen. Das Gleiche gilt natürlich auch für die Anzahl richtig erzeugter Kombinationen, die nicht zwangsläufig die richtige Agentenmenge produzieren. Um diese Gewichte bewerten zu können, werden zwei zusätzliche Indikatorensätze definiert.

Abbildung 2.2: Simulierte Observationsindikatoren und Populationsindikatoren



Die nächsten beiden Diagramme sind keine Venn-Diagramme, sie dienen vielmehr der Visualisierung der Herkunft der neuen Indikatoren. Das erste Diagramm (Abbildung 2.2 a) stellt das Ergebnis der simulierten Menge von Observationen dar, d.h. die synthetische Population. Diese wird in 3 Regionen unterteilt. Die erste Region beinhaltet die Menge aller erzeugten Observationen, deren Attributenkombination in der realen Population nicht vorhanden ist. Ihre Anzahl wird durch den Indikator *new.obs.sim* quantifiziert. Diese Menge kommt zustande, weil der Algorithmus den „falschen“ Attributenkombinationen, die in der realen Population nicht existieren, Gewichte grösser null zuweist. Die zweite Region (*match.in.obs.sim*) spiegelt die Anzahl simulierter Observationen wieder, deren Attributenkombination in der Stichprobe enthalten ist. N.B. Ihre Anzahl ist nicht gleichzusetzen mit der Anzahl von Observationen, die sich in der Schnittmenge zwischen der realen und der synthetischen Population befinden. Die dritte Region (*match.out.obs.sim*) enthält die Anzahl vom Algorithmus erzeugter Observationen, die auch in der realen Population vorhanden sind, auch hier nicht zu verwechseln mit der Anzahl in der Schnittmenge. Das zweite Diagramm (Abbildung 2.2 b) ist analog zum ersten mit dem Unterschied, dass es auf die reale Population bezogen ist. Dabei bedeuten die Indikatoren *missing.in.obs.pop* und *missing.out.obs.pop* die Anzahl nicht vom Modell erzeugten Observationen, deren Attributenkombinationen in der Stichprobe enthalten sind bzw. fehlen. Die Indikatoren *match.in.obs.pop* bzw. *match.out.obs.pop* spiegeln die Anzahl Observationen wieder, deren Attributenkombination in der Stichprobe präsent ist bzw. fehlt und vom Algorithmus in Observationen konvertiert wird.

Abbildung 2.3: Vergleichsindikatoren

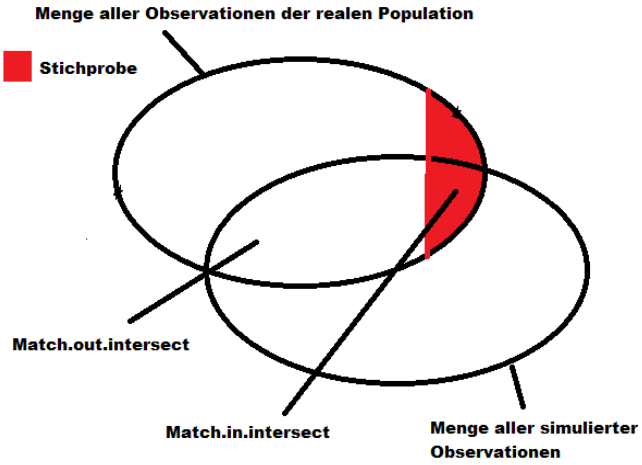


Abbildung 2.3 ist ein Venn-Diagramm. Um die Performance des Algorithmus besser darzustellen, werden die Indikatoren *match.in.intersect und match.out.intersect* definiert. Diese werden durch die Aufsummierung aller jeweils kleineren Frequenzen der Observationen des Typs match.in zwischen der realen und der simulierten Population erhalten. Sie werden durch *match.intersect* zusammengefasst und geben die Grösse der Schnittmenge zwischen der synthetischen Population und der realen wieder.

Zuletzt werden noch sechs neue Indikatoren hinzugefügt, mit denen Fehler quantifiziert werden können. *match.in.mae* bzw. *match.out.mae* geben den absoluten Fehler wieder und *match.in.mrae* bzw. *match.out.mrae* den relativen Fehler. Alle vier Indikatoren beziehen sich auf Observationen mit einer Attributenkombination, die in der synthetischen und in der realen Population gleich ist (*match.in/out*), wobei deren Anzahl mit *PopFreq* (in der realen Population) und *SimFreq* (in der synthetischen Population) angegeben wird.

Match.in.mae= $\sum_{match.in} |PopFreq - SimFreq|$

Match.out.mae= $\sum_{match.out} |PopFreq - SimFreq|$

Match.in.mrae= $\sum_{match.in} \frac{|PopFreq - SimFreq|}{SimFreq}$

Match.out.mrae= $\sum_{match.out} \frac{|PopFreq - SimFreq|}{SimFreq}$

Mit

$Match.\,in\ bzw.\,out$ = Agenten deren Attributenkombination es in der realen Population gibt
$PopFreq$ = relative Häufigkeit einer Observation in der realen Population
$SimFreq$ = erwartete, relative Häufigkeit in der synthetischen Population

Die „Fehlerindikatoren“ werden durch *match.in.kl* und *match.out.kl* vervollständigt. Diese geben die Kullback-Leibler-Divergenz an, welche ein Mass für die Unterschiedlichkeit zweier Wahrscheinlichkeitsverteilungen liefert. Sie wird verwendet, um auszudrücken, wie gut die

Verteilungsfunktion der Personen der realen Population durch die des Modells beschrieben wird. Auch hier werden nur Observationen berücksichtigt, die eine Attributenkombination aufweisen, die es sowohl in der realen als auch in der synthetischen Population gibt. Die Auftretenswahrscheinlichkeit einer Observation in der realen bzw. synthetischen Population wird durch *PopProb* bzw. *SimProb* angegeben.

$$match.in.kl = \sum_{match.in} PopProb * ln\left(\frac{PopProb}{SimProb}\right)$$

$$match.out.kl = \sum_{match.out} PopProb * ln\left(\frac{PopProb}{SimProb}\right)$$

Mit

$Match.in\ bzw.out$ = Agenten, deren Attributenkombination es in der realen Population gibt
$PopProb$ = Auftretenswahrscheinlichkeit der Observation in der realen Population
$ProbSim$ = Auftretenswahrscheinlichkeit des Agenten in der synthetischen Population

2.4.6 Die Indikatorensätze

Da die Erzeugung einer synthetischen Population für jede Stichprobe in allen Szenarien zu zeitraubend wäre, wird der Algorithmus nicht in allen getestet. Es wird mit zwei vorberechneten Indikatorensätzen gearbeitet. Diese beinhalten die Werte jedes Indikators für bestimmte Parameterkombinationen in vorbestimmten Szenarien und das für alle 100 Stichproben.

Indicators.1: beinhaltet alle Indikatorenwerte für jede Parameterkombination in den Szenarien r ∈ [0.01; 0.02; 0.05; 0.1; 0.2] und ncols=9

Indicators.2: beinhaltet alle Indikatorenwerte für jede Parameterkombination mit Weighted=FALSE für die Szenarien r ∈ [0.01; 0.02; 0.05; 0.1; 0.2] und ncols ∈ [3; 12]

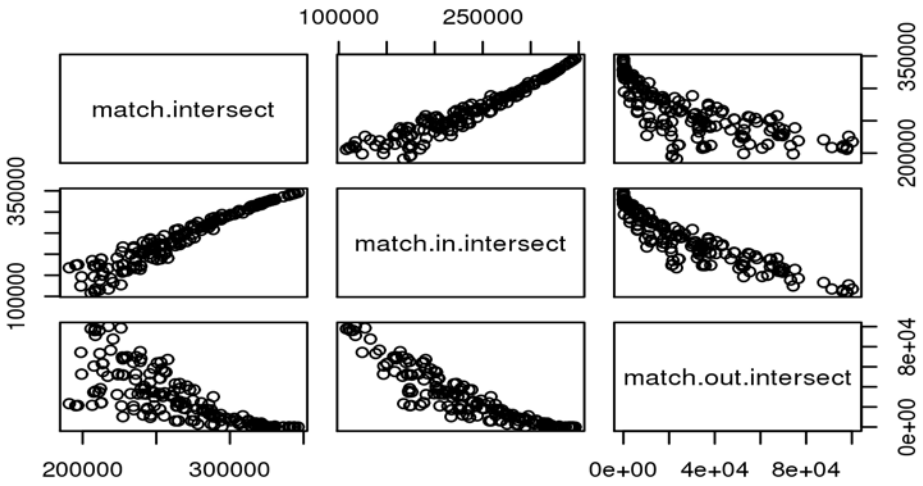
2.5 Interdependenz der Indikatoren

Der Zusammenhang zwischen den verschiedenen Indikatoren wird anhand von Korrelationsplots ermittelt. Diese bestehen aus 200 zufälligen Ziehungen aus dem Indikatorensatz **indicators.2.** Der Fokus liegt dabei auf der Korrelation des Indikator match.intersect mit den anderen Indikatoren. *Match.intersect* entsteht aus der Summe der Indikatoren *match.in.intersect* und *match.out.intersect* und spiegelt die Menge von korrekt abgebildeten Personen auf synthetische Agenten wieder. Je mehr reale Personen auf einen Agenten, der die gleiche Attributenkombination aufweist, abgebildet werden können, desto mehr gleicht die synthetische Population der realen und ist desto qualitativ hochwertiger einzustufen. *Match.intersect* ist ausserdem der einzige Indikator, mit dem es alleine möglich ist, eine perfekte, synthetische Population zu identifizieren, was mit den Fehlerindikatoren alleine nicht geleistet werden kann. Würden diese alle den Wert null annehmen, würde dies nicht zwangsläufig bedeuten, dass die synthetische Population identisch mit der realen ist. Dies kann mittels der vereinfachten Betrachtung des hypothetischen Szenarios verstanden werden, in dem versucht wird, eine reale Population, die aus drei verschiedenen Personen besteht, mittels einer Stichprobe, die nur eine enthält, zu rekonstruieren. Die erzeugte synthetische Population besteht aus zwei Agenten des Typs match (eine *match.in.obs.sim* und eine

match.out.obs.sim) und einem des Typs new (*new.obs.sim*). Alle Fehlerindikatoren würden in diesem Fall den Wert null aufweisen, auch wenn die künstliche Population nur zu zwei Drittel der realen gleicht. *Match.intersect* würde in diesem Fall den Wert zwei annehmen und da dieser Wert nicht der totalen Anzahl Personen der realen Population entspricht, fällt die Differenz zwischen den beiden Populationen auf.

Abbildung 2.4 verdeutlicht die Korrelation zwischen den Indikatoren *match.intersect*, *match.in.intersect* und *match.out.intersect*. Es ist deutlich zu erkennen, dass *match.in.intersect* und *match.out.intersect* eine negative Korrelation aufweisen. Die Vergrösserung der Gesamtmenge von richtig abgebildeten Personen (*match.intersect*) ist positiv mit *match.in.intersect* korreliert aber negativ zu *match.out.intersect*. Anhand dieser Betrachtungsweise ist ein Konflikt zwischen den richtig simulierten Observationen, deren Attributenkombination in der Stichprobe vorhanden ist, und denen, bei denen dies nicht der Fall ist, zu erkennen.

Abbildung 2.4: Korrelation zwischen den match.intersect Indikatoren



Diese Korrelation ist bei allen Szenarien zu erkennen. Abbildung 6.1a/b (Anhang) zeigt dies für eine grosse Anzahl von berücksichtigten Attributen bei einer grossen Stichprobe (a) und bei einer kleinen (b). Dasselbe gilt für eine niedrige Attributenanzahl (Abbildung 6.1 c/d im Anhang) bei r=0.1 (c) und r=0.01 (d)).

Abbildungen 2.5 a: Korrelation zwischen match.out.intersect, match.out.kl.pop.sim und match.out mrae
b: Korrelation zwischen match.in.intersect, match.in.kl.pop.sim und match.in.mrae

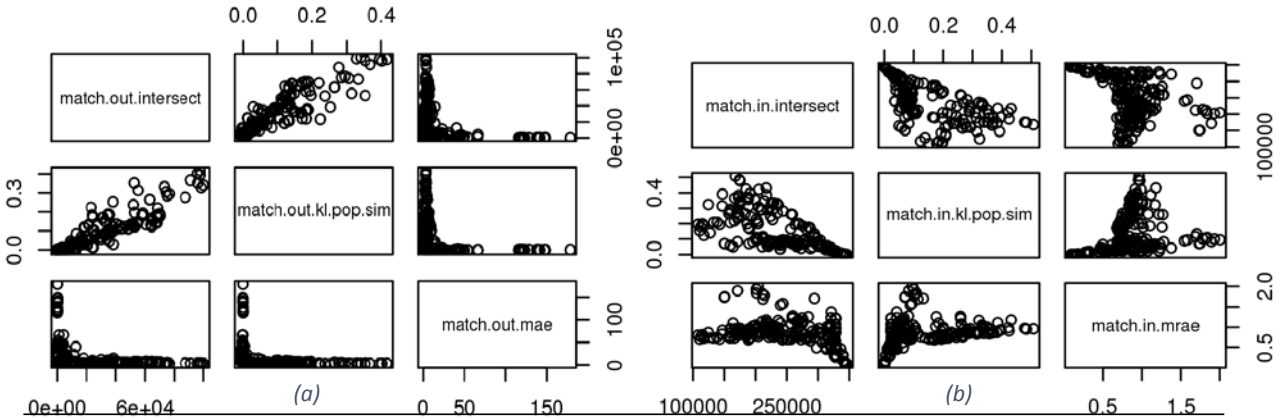
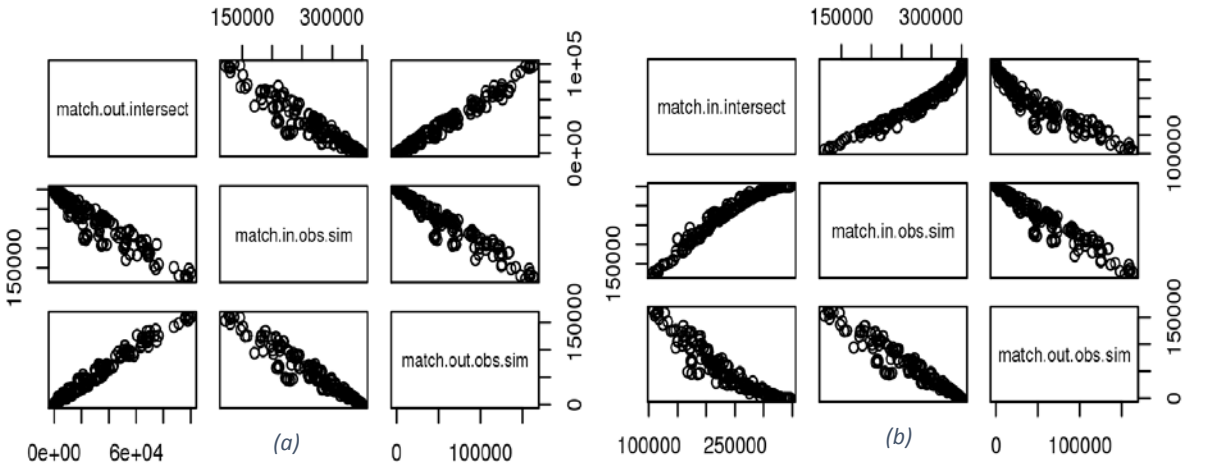


Abbildung 2.5 a/b zeigt die Korrelation der Indikatoren *match.in/out.intersect*, *match.in/out.mrae* und *match.in/out.kl*. Man kann erkennen, dass *match.in/out.intersect* und *match.in/out.kl* deutliche Korrelationstendenzen aufweisen. Steigt der Wert des Indikators *match.in/out.intersect*, sinkt der der entsprechenden K.L.-Divergenz. Die fehlende Korrelation zwischen *match.in/out.mrae* und den anderen beiden Indikatoren könnte in der starken Zunahme des absoluten relativen Fehlers bei einer grösseren Frequenz von simulierten Agenten des Typs match, im Gegensatz zu den realen liegen. Das Gleiche passiert bei der K.L.-Divergenz. In diesem Fall dämpft aber der Logarithmus die starke Zunahme. Auch die Verteilung der Differenzen zwischen den Frequenzen der Attributenkombinationen spielt eine tragende Rolle. Eine grosse Differenz zwischen einer niedrigen Personen-Frequenz mit einer bestimmten Attributenkombination bei einer hohen synthetischen Agentenfrequenz mit derselben Attributenkombination führt dazu, dass der absolute relative Fehler stark zunimmt. Die gleichen Aussagen können in den verschiedenen Szenarien bei grosser und kleiner Attributen Berücksichtigung mit kleinen und grossen Stichproben gemacht werden (Abbildungen 6.2 und 6.3 im Anhang).

Abbildung 2.6 a: Korrelation zwischen match.in.intersect, match.in.obs.sim und match.out.obs.sim
b: Korrelation zwischen match.out.intersect, match.in.obs.sim und match.out.obs.sim



Betrachtet man den Zusammenhang der Indikatoren *match.in/out.intersect* (Abbildung 2.6 a/b) mit der Anzahl erzeugter Observationen, deren Attributenkombination in der realen Population zu finden ist, so sind klare Tendenzen zu erkennen. Die Erzeugung einer grossen Menge von

Observationen, deren Attributenkombination aus der Stichprobe stammt (*match.in.obs.sim*), hat zur Folge, dass der Wert des Indikators *match.in.intersect* steigt. Dasselbe gilt für *match.out.obs.sim* und *match.out.intersect*. Es wird wieder der Konflikt zwischen den zwei Regionen match.in und match.out deutlich bemerkbar. Aus der Vergrösserung der einen folgt eine Verkleinerung der anderen. Auch hier gilt das, was vorher zu den verschiedenen Szenarien gesagt wurde (Abbildungen 6.4 und 6.5 im Anhang).

Abbildung 2.7 a: Korrelation zwischen match.in.intersect und new.obs.sim
b: Korrelation zwischen match.out.intersect und new.obs.sim

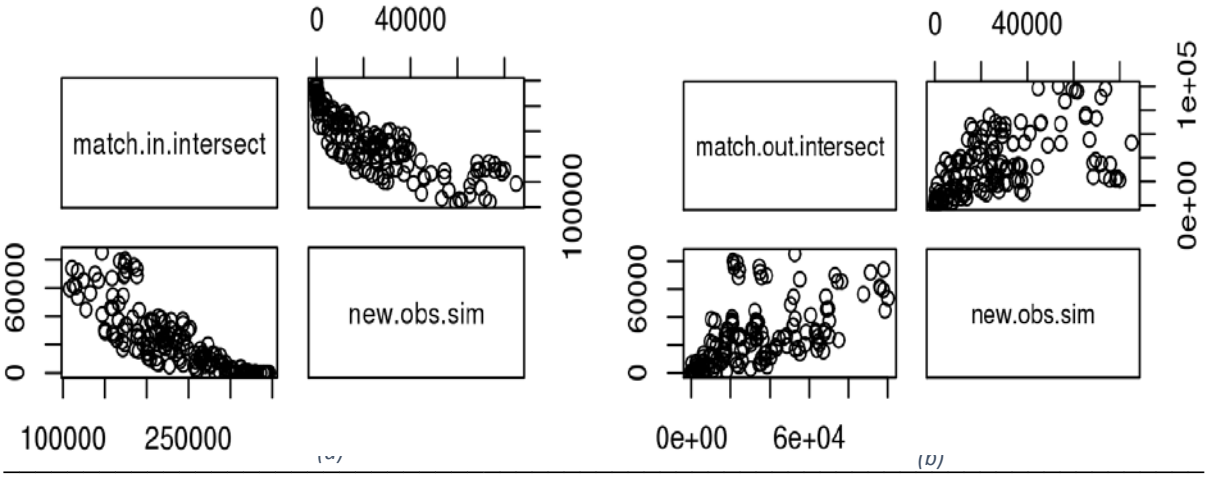


Abbildung 2.7 a/b zeigt, dass sich eine grössere Anzahl neu simulierter Observationen (*new.obs.sim*) positiv auf den Indikator *match.out.intersect* auswirkt und negativ auf den Indikator *match.in.Intersect*. Dies kann erklärt werden, indem berücksichtigt wird, dass die Wahrscheinlichkeit, einen oder mehrere Agenten zu erzeugen, deren Attributenkombination nicht aus der Stichprobe stammt, aber in der realen Population präsent ist, deutlich steigt, wenn eine grössere Menge von neuen Agenten, deren Attributenkombination nicht in der Stichprobe zu finden ist, erzeugt wird. Deshalb werden bei einer grösseren Anzahl von *new.obs.sim* mehr Treffer gelandet (*match.out.intersect*). Es ist offensichtlich, dass sich eine grosse Anzahl Agenten des Typs new negativ auf *match.in.intersect* auswirkt. Da die reale Population zum grössten Teil aus Personen besteht, deren Attributenkombination in der Stichprobe landet, wird eine Vergrösserung der Menge *new.obs.sim* zum Ungunsten der Menge *match.in.obs.sim* vorgenommen. . Dies gilt für die verschiedenen Szenarien (Abbildung 6.6).

Abbildung 2.8: Korrelation zwischen match.in.obs.sim, match.out.obs.sim und new.combs.sim

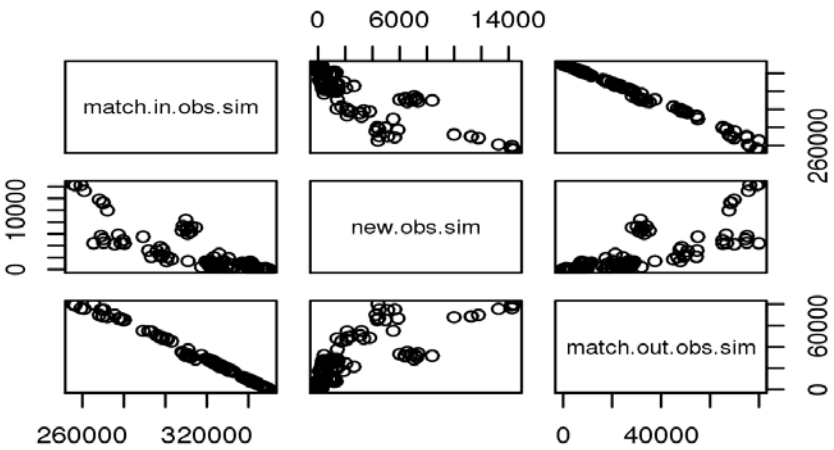


Abbildung 2.8 zeigt die positive Korrelation zwischen *new.obs.sim* und *match.out.obs.sim* und die negative zwischen *match.in.obs.sim match.out.obs.sim* und *new.obs.sim*. Man erkennt, dass die beiden Mengen von neu erzeugten Agenten, deren Attributenkombination nicht in der Stichprobe zu finden ist (new bzw. match.out), gleichzeitig wachsen oder schrumpfen. Aus einer simultanen Vergrösserung dieser beiden Teilmengen folgt logischerweise eine Verkleinerung der Menge von Observationen, deren Attributenkombination in der Stichprobe enthalten ist, da die Summe dieser drei Teilmengen immer den gleichen Wert aufweist, und zwar den der Anzahl Personen der gesamten realen Population. Auch dies ist in allen Szenarien zu beobachten (Abbildung 6.7).

Auswertung der Korrelationen

Die Qualität der simulierten Populationen wird im Folgenden hauptsächlich anhand des Indikators *match.intersect* gemessen. Dieser Indikator drückt in einfacher und anschaulicher Weise die Grösse der übereinstimmenden Menge zwischen der simulierten und der realen Population aus, das heisst die Anzahl korrekt abgebildeter Personen auf synthetische Agenten. Die Aussagekraft dieses Indikators wird durch die Korrelation mit der K.L.-Divergenz unterstrichen, auch wenn eine gewisse Streuung zu berücksichtigen ist. Dies bedeutet, dass eine bessere Approximation der multivariaten Wahrscheinlichkeitsverteilung zu einer grösseren Anzahl exakt reproduzierter Agenten führt. Der absolute relative Fehler wird nicht aus den Augen verloren, da keine eindeutige Korrelation mit *match.intersect* festgestellt werden kann. Aus diesen ersten Grafiken gewinnt man den Eindruck, dass die beiden Teilmengen match.in und match.out nicht zusammen maximiert werden können, also dass das Wachstum der einen zu einer Schrumpfung der anderen führt. Dies wird anhand der Abbildung 2.4 veranschaulicht. Einer grösseren „in of sample"-Menge folgt eine geringere „out of sample" Menge. Mit anderen Worten: Werden mehr Agenten synthetisiert, deren Attributenkombination aus der Stichprobe entnommen wurde, so werden weniger synthetisiert, deren Kombination in der realen Population vorhanden ist, aber nicht in der Stichprobe liegt. Dies wirkt sich direkt auf die Indikatoren *match.in.intersect* und *match.out.intersect* aus, die in unmittelbaren Zusammenhang zu der gerade erläuterten Menge stehen. Ein grösserer Anteil von Observationen des Typs match.in führt zu einer Vergrösserung der Überlappungsregion *match.in.intesect* zu Ungunsten derer von *match.out.intersect*. Das Umgekehrte ist bei einer Erhöhung der Observationen des Typs match.out festzustellen (Abbildung 2.6). Abbildung 2.7 zeigt die Auswirkung der simulierten Observationen, die keiner Person aus der realen Population gleichen. Es ist offensichtlich, dass einer Vergrösserung dieser Menge auch eine der der Observationen des Typs match.out folgt und dass diese sich parasitär im Hinblick auf die Menge des Typs match.in verhält. Die vielleicht wichtigste Feststellung ergibt sich aus Abbildung 2.7b. Diese zeigt die Tendenz, dass eine steigende Anzahl neuer Observationen (im Vergleich zur realen Population) zu einer Vergrösserung der Region *match.out.intersect* führt. Dies kann durch eine einfache Wahrscheinlichkeits-betrachtung erklärt werden. Steigt die Anzahl simulierter Observationen, deren Attributenkombination nicht in der Stichprobe liegt, so steigt auch Wahrscheinlichkeit, dass eine des Typs match.intersect erzeugt wird. Abbildung 2.7a zeigt die negative Wirkung dieser Observationen auf match.in.intersect, die zu dessen Ungunsten wachsen. Es bleibt nun die Frage offen, wie dieser Konflikt angegangen werden soll, d.h welche der beiden Überlappungsregionen bevorzugt werden sollte. Folgt man der Erkenntnis aus Abbildung, 2.4 wird die Frage mit der Menge match.in beantwortet, da ihre Vergrösserung derjenigen, einer der Gesamtmenge *match.intersect* folgt, während einer

Vergrösserung der match.out Menge zu einer Verkleinerung der Gesamtüberlappungsregion führt. Man sollte aber vorsichtig sein, da sich diese Verhältnisse mit einer Verringerung der Stichprobengrösse und einer steigenden Attributenanzahl zu Gunsten von match.out verschieben könnten. Bei grossen Stichproben und wenigen Attributen ist offensichtlich, dass der mit Abstand grösste Teil der Population einfach durch die Expansion der Stichprobe erhalten werden kann, da die meisten Attributenkombinationen der Population in der Stichprobe landen. Steigt hingegen die Anzahl Attribute und verkleinert sich die Stichprobengrösse, so ist dies nicht mehr der Fall, da die Stichprobe einen viel niedrigeren Informationsgehalt bezüglich der Gesamtpopulation aufweist. Dies ist unter dem Prinzip des Fluches der Dimensionalität bekannt, welches die extreme Schrumpfung des Verhältnisses zwischen dem Volumen einer Menge und dem Volumen des Raumes der diese Menge beinhaltet bei Addition von weiteren Dimensionen erläutert. Das heisst, dass die Attributenkombinationsanzahl einer Stichprobe im Verhältnis zur gesamten Anzahl möglicher Kombinationen sehr klein wird, wenn viele Attribute berücksichtigt werden. Dies wird anhand Abbildung 2.9 veranschaulicht.

Abbildung 2.9 a: Häufigkeitsverteilung der Kombinationen in der Population bei 13 Attributen
b: Häufigkeitsverteilung der Kombinationen in der Population bei 5 Attributen

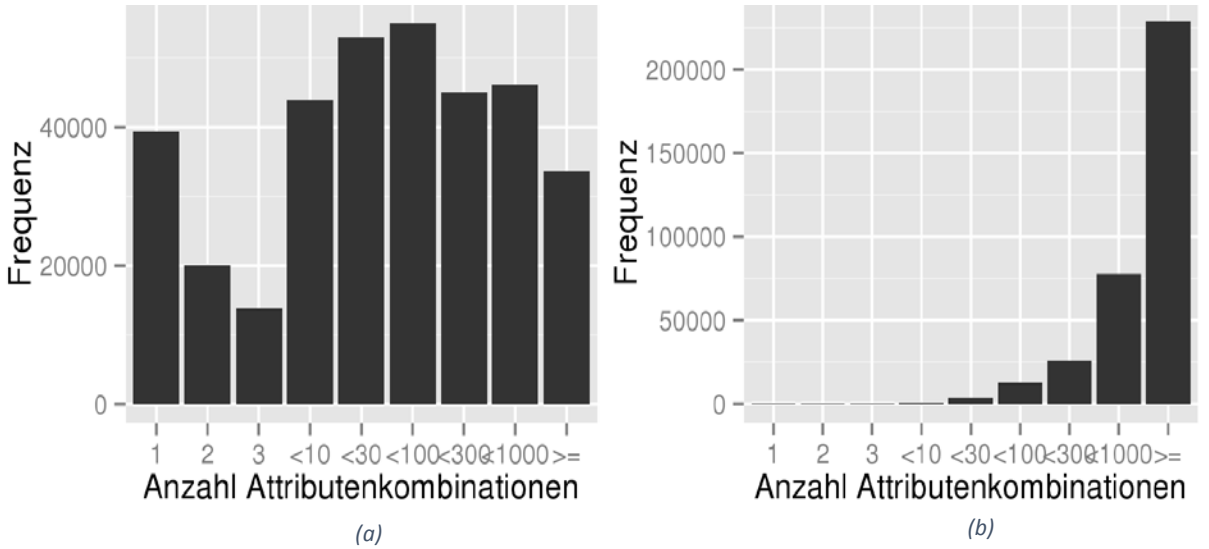


Abbildung 2.9 zeigt die Häufigkeit verschiedener Attributenkombinationen bei 13 und 5 berücksichtigten Attributen der Population des Experimentes. Jedes Histogramm zeigt die Auftretenshäufigkeit von Kombinationen, die es in der Anzahl, die von der x-Achse abgelesen werden kann, in der Population gibt. Zum Beispiel gibt es bei 13 berücksichtigten Attributen fast 40000 verschiedene Attributenkombinationen die einzigartig sind (Abbildung 2.9 a). Man sieht, dass bei 13 Attributen die Histogramme im linken Teil des Diagramms stark repräsentiert sind, was einer grossen Anzahl von rar auftretenden Attributenkombinationen entspricht. Bei 5 Attributen ist dies nicht der Fall. Dort tritt jede Kombination in einer grossen Menge auf (Abbildung 2.9 b).

3 Auswirkung der Parameter auf die Indikatoren

3.1 Das lineare Modell

Um eine erste Übersicht über die Grössenordnung des Einflusses der verschiedenen Parameter auf die Qualität der synthetischen Population zu bekommen, wird eine erste Analyse anhand eines linearen Modells durchgeführt. Dieses basiert auf der Methode der kleinsten Quadrate und führt die Schätzung anhand der Datensätze indicators.1 und Indikators.2 separat aus.

Tabelle 3.1 Schätzung des linearen Modells (indicators.2)

		match.intersect	match.in.intersect	match.out.intersect	new.obs.sim	match.in.obs.sim	match.out.obs.sim
cp.class.neg	Wert	390900.00	437400.00	-46490.00	-35370.00	465300.00	-80183.62
	sd	317.10	439.00	242.50	215.70	508.20	415.97
	Signifikanz	<2e-16	<2e-16	< 2e-16	<2e-16	<2e-16	< 2e-16
cp.class.zero	Wert	385500.00	423800.00	-38290.00	-30380.00	447400.00	-67199.05
	sd	317.10	439.00	242.50	215.70	508.20	415.97
	Signifikanz	<2e-16	<2e-16	< 2e-16	<2e-16	<2e-16	< 2e-16
cp.class.pos	Wert	368100.00	401200.00	-33170.00	-19180.00	422800.00	-53866.51
	sd	332.60	460.40	254.40	226.20	533.00	436.29
	Signifikanz	<2e-16	<2e-16	< 2e-16	<2e-16	<2e-16	< 2e-16
cp.nonneg	Wert	-1056000.00	-981000.00	-75360.00	430800.00	-715500.00	284678.76
	sd	13130.00	18180.00	10040.00	8932.00	21050.00	17227.56
	Signifikanz	<2e-16	<2e-16	0.00	<2e-16	<2e-16	< 2e-16
weighted TRUE	Wert	NA	NA	NA	NA	NA	NA
	sd	NA	NA	NA	NA	NA	NA
	Signifikanz	NA	NA	NA	NA	NA	NA
MM	Wert	2346.00	2084.00	261.70	-5069.00	4113.00	956.30
	sd	126.90	175.70	97.03	86.29	203.30	166.43
	Signifikanz	<2e-16	<2e-16	0.01	<2e-16	<2e-16	0.00
ri	Wert	-177.90	-559.50	381.60	-3.73	-628.10	631.83
	sd	1.80	2.50	1.38	1.23	2.89	2.37
	Signifikanz	<2e-16	<2e-16	< 2e-16	0.00	<2e-16	< 2e-16
ncols	Wert	-12980.00	-19430.00	6451.00	6957.00	-17840.00	10880.67
	sd	22.09	30.58	16.89	15.02	35.39	28.97
	Signifikanz	<2e-16	<2e-16	< 2e-16	<2e-16	<2e-16	< 2e-16
i	Wert	-1.36	-0.75	-0.62	0.75	0.27	-1.01
	sd	2.20	3.04	1.68	1.50	3.52	2.88
	Signifikanz	0.54	0.81	0.71	0.62	0.94	0.73

Die Werte von Tabelle 3.1 sind nicht realistisch, da sie aus einer linearen Schätzung stammen, was durch ihre Signifikanz bestätigt wird. Diese ersten Resultate sind aber ein wichtiges Werkzeug, um die Tendenzen der Steuerungsparameter des Algorithmus zu verstehen. Die Tabelle zeigt den geschätzten Anteil jedes Parameters am betrachteten Indikator. Die fünf verschiedenen cp Werte werden in drei Klassen aufgeteilt. Class.neg entspricht cp=-1, class.zero cp=0 und class.pos cp=0.005, cp=0.01 und cp=0.02. cp.nonneg ist die Differenz zwischen den Ergebnissen zwischen den cps der Klasse class.pos. Die Zeilen MM, weighted TRUE, ri, ncols, i zeigen die Auswirkung auf den betrachteten Indikator bei einer Erhöhung des Parameters auf den nächsten Wert des vordefinierten Intervalls bzw. einer Änderung von FALSE zu TRUE. Der Parameter ri ist gleich 1/r und bei einer Änderung dieses Parameter muss der Wert des linearen Modells mit 1/r multipliziert werden.

Man kann erkennen, dass der cp Parameter der mit Abstand einflussreichste ist. Dies kann an den grösseren Werten in der Tabelle abgelesen werden. Niedrige Werte dieses Parameters führen im Vergleich zur realen Population zur global ähnlichsten, simulierten Population, was bei der Betrachtung des Indikators *match.intersect* offensichtlich wird. Die Indikatoren *match.in.intersect* und *match.out.intersect* verhalten sich bei einer Variation des cp-Wertes invers proportional zu einander. Einem niedrigen cp-Wert folgen grosse Werte des Indikators *match.in.intersect* und kleine des Indikators *match.out.intersect*. Das Umgekehrte passiert bei hohen cp-Werten. Der Parameter MM hat im Vergleich zu cp nur einen geringen Einfluss auf den Indikator *match.intersect*. Aus der Auswertung der Resultate des linearen Modells ist auch zu erkennen, dass die Stichprobe fast keinen Einfluss auf die betrachteten Indikatoren hat.

Tabelle 3.2: Schätzung des linearen Modells (indicators.1)

		match.intersect	match.in.intersect	match.out.intersect	new.obs.sim	match.in.obs.sim	match.out.obs.sim
	Wert	285000.00	280000.00	4942.00	18478.85	325100.00	6225.00
cp.class.neg	sd	252.90	258.20	152.80	143.67	311.00	265.20
	Signifikanz	0.00	<2e-16	< 2e-16	0.00	<2e-16	<2e-16
	Wert	279100.00	264200.00	14860.00	24701.36	302500.00	22600.00
cp.class.zero	sd	252.90	258.20	152.80	143.67	311.00	265.20
	Signifikanz	0.00	<2e-16	< 2e-16	0.00	<2e-16	<2e-16
	Wert	253600.00	233400.00	20270.00	36235.64	273000.00	40540.00
cp.class.pos	sd	278.00	283.90	168.00	157.96	342.00	291.60
	Signifikanz	0.00	<2e-16	< 2e-16	0.00	<2e-16	<2e-16
	Wert	-1298000.00	-1232000.00	-66050.00	496293.81	-924900.00	428600.00
cp.nonneg	sd	15130.00	15450.00	9145.00	8597.01	18610.00	15870.00
	Signifikanz	0.00	<2e-16	0.00	0.00	<2e-16	<2e-16
	Wert	1007.00	-327.90	1335.00	377.72	-2178.00	1800.00
weighted TRUE	sd	146.20	149.30	88.35	83.06	179.80	153.30
	Signifikanz	0.00	0.03	< 2e-16	0.00	<2e-16	<2e-16
	Wert	2149.00	3378.00	-1229.00	-5962.70	7672.00	-1710.00
MM	sd	65.37	66.76	39.51	37.14	80.42	68.57
	Signifikanz	0.00	<2e-16	< 2e-16	0.00	<2e-16	<2e-16
	Wert	-217.90	-693.80	475.90	-32.87	-766.20	799.10
ri	sd	2.08	2.12	1.26	1.18	2.56	2.18
	Signifikanz	0.00	<2e-16	< 2e-16	0.00	<2e-16	<2e-16
	Wert	NA	NA	NA	NA	NA	NA
ncols	sd	NA	NA	NA	NA	NA	NA
	Signifikanz	NA	NA	NA	NA	NA	NA
	Wert	-2.93	-2.74	-0.20	1.36	-0.59	-0.77
i	sd	2.53	2.59	1.53	1.44	3.12	2.66
	Signifikanz	0.25	0.29	0.90	0.34	0.85	0.77

Tabelle 3.2 zeigt die gleichen Tendenzen wie Tabelle 3.1. Man sieht, dass auch weighted eine untergeordnete Rolle im Hinblick auf die Indikatorenfamilie match.intersect hat.

3.2 Erste Erkenntnisse

Nach einer ersten Betrachtung der Indikatorensätze indicators.1 und indicators.2, wird Folgendes festgestellt:

1. *Missing.in.combs* und *missing.in.obs.pop* weisen immer den Wert 0 auf.

2. Alle Parameterkombinationen, die sich nur mittels MM=0 und MM=1 unterscheiden, weisen dieselben Indikatorenwerte auf.

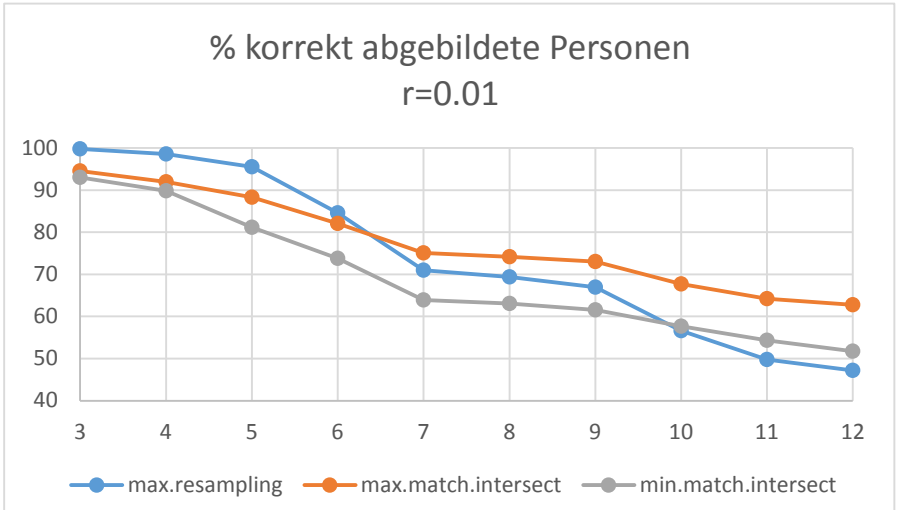
Daraus kann Folgendes interpretiert werden:

1. Der Algorithmus erkennt jede, in der Stichprobe vorhandene Attributenkombination und materialisiert sie in Form von synthetischen Agenten in der simulierten Population.

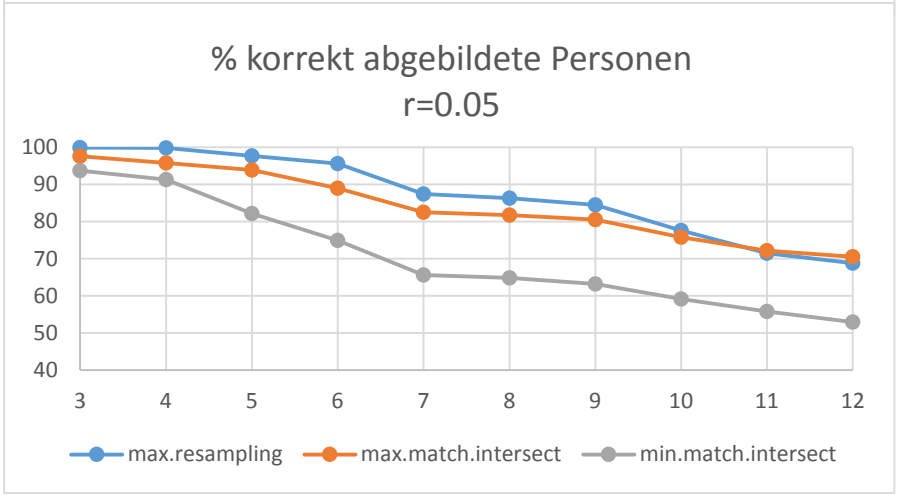
2. Da in jeder vom Algorithmus erzeugten Population immer Agenten mit jeder Attributenkombination vorhanden sind, die der jedes Agenten der Stichprobe entsprechen, steht fest, dass *match.in.combs* nur von der Stichprobengrösse und der Anzahl berücksichtigter Attribute abhängt.

3. Das identische Verhalten der Parameterkombinationen mit MM=0 und MM=1 kann erklärt werden, indem die Wahrscheinlichkeit in Betracht gezogen wird, dass in den Attributenkombinationen, die in der Stichprobe enthalten sind, alle Attributenkategorien mindestens einmal vorkommen.

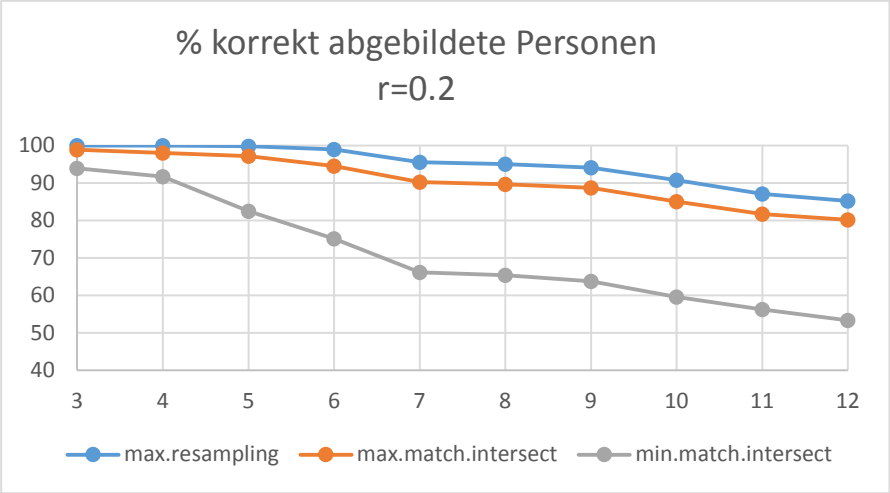
Abbildung 3.1 a, b und c % Match.intersect für r ∈ [0.01, 0.05, 0.1]



(a)



(b)



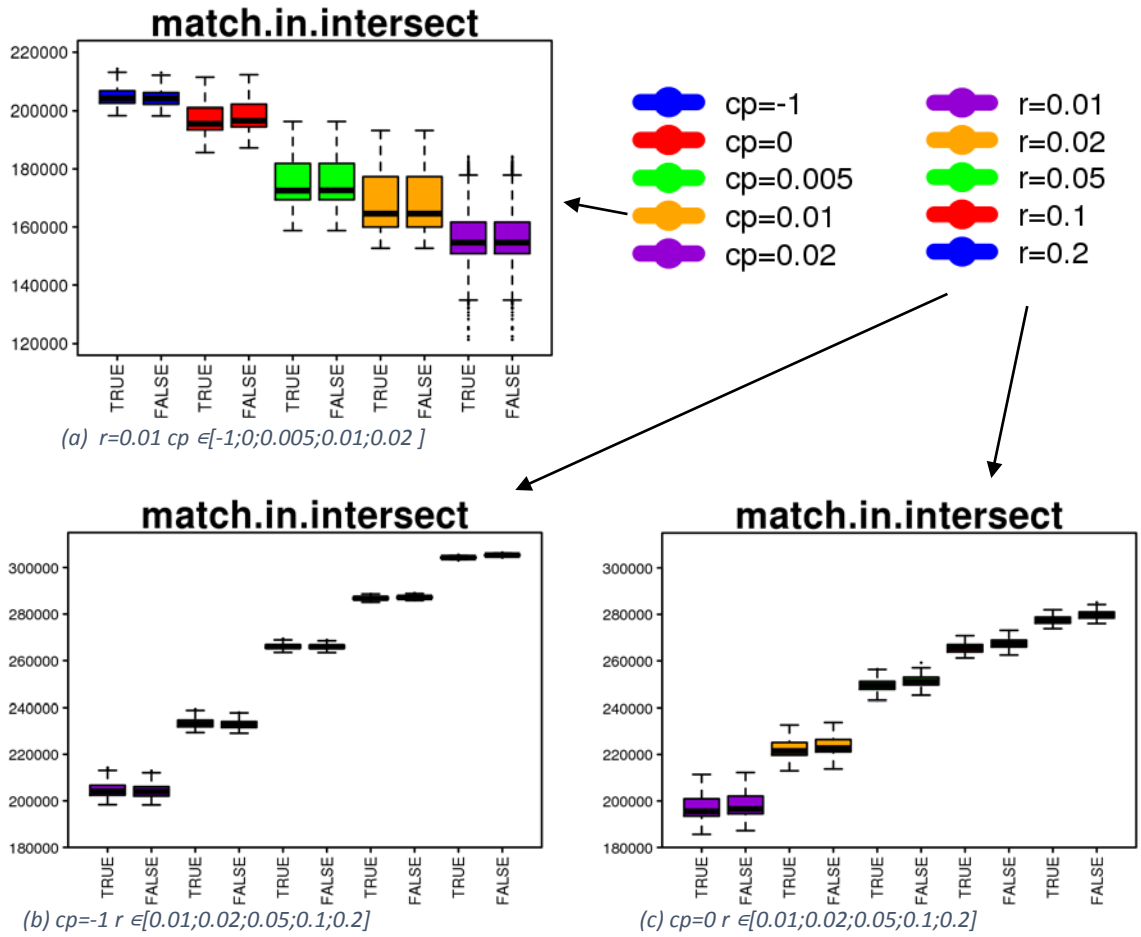
(c)

Um eine erste Idee der Performance des Algorithmus zu bekommen, wird die Anzahl korrekt abgebildeter Personen auf synthetische Agenten (*match.intersect*) durch die Anzahl Personen der gesamten Population dividiert. Dies wird anhand des Indikatoren Satzes indicators.2 und für drei verschiedene Stichprobenraten r=0.01, 0.05 und 0.2 gemacht (Abbildung 3.1). Es wird jeweils das Resultat der besten und der schlechtesten Parameterkombination eingetragen und mit *max.resampling* verglichen. *Max.resampling* ist das Ergebnis eines perfekten Resampling-Algorithmus, der jede Observation der Stichprobe so expandiert, dass deren Frequenz in der synthetischen Population grösser oder gleich der der realen Population ist. Es fällt auf, dass der Algorithmus mit einer geringen Stichprobengrösse und steigender Anzahl Attribute bessere Ergebnisse liefert als das perfekte Resampling Modell und das sogar mit der schlechtesten Parameterkombination. Diese einfache Betrachtung verdeutlicht das Gewicht der Parameterkombination, mit der der Algorithmus gesteuert wird, auf die resultierende synthetische Population, welche man aus der vertikalen Differenz der Linien *max.match.intersect* und *min.match.intersect* erkennen kann.

3.3 Der Gewichtungsparameter Weighted

Mit diesem Parameter wird entschieden, wie die Observationsmenge behandelt wird. Bei weighted=TRUE werden gleiche Observationen, d.h. solche mit gleicher Attributenkombination zu einer einzigen zusammengefasst und mit einem Gewicht versehen. Bei FALSE werden sie einzeln behandelt. Vom theoretischen Ansatz dürfte kein Unterschied festzustellen sein, was aber nicht der Fall ist. Der Parameter weighted wird anhand des Indikatorensatzes indicators.1 untersucht. Für die Indikatoren *match.intesect*, *match.out.intersect* und *match.intersect* werden für die Stichprobengrössen r=0.01 10 Boxplots erstellt, die sich anhand des cp-Parameters und des Gewichts ∈ [TRUE, FALSE] unterscheiden. Der Unterschied zwischen den Parameterkombinationen, die sich nur anhand weighted=TRUE oder FALSE unterscheiden, ist nur für cp-Werte ∈ [-1, 0] zu erkennen (siehe Abbildungen 3.2 a und 3.3 a und d). Deswegen werden für r ∈ [0.02, 0.05, 0.1, 0.2] nur die cp ∈[-1, 0] analysiert. Die Analyse des Parameters weighted im Hinblick auf die anderen Indikatoren ist im Anhang zu finden.

Abbilduna 3.2 a, b und c



(a) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

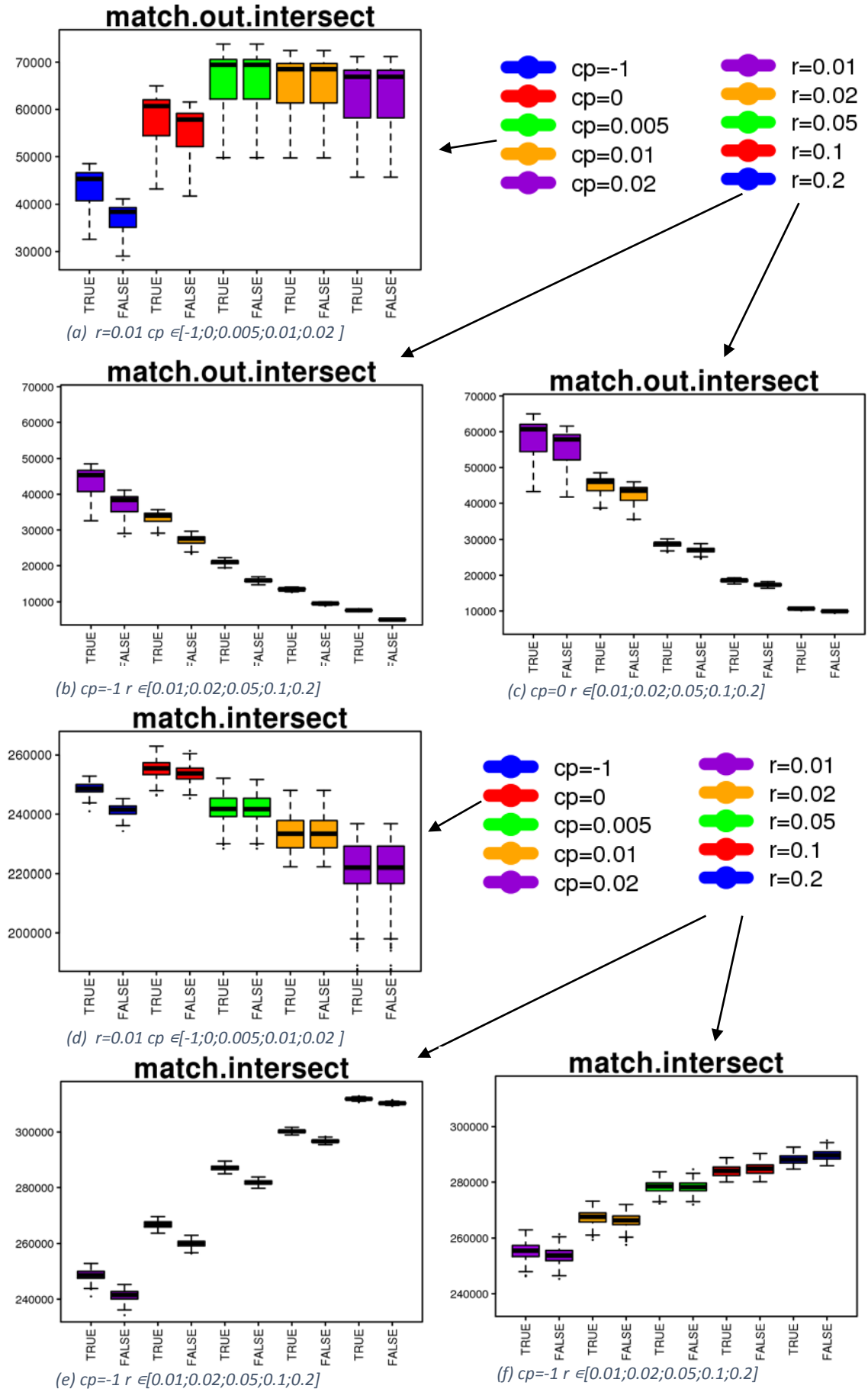
(b) cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

(c) cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

Die synthetische Population, die mittels weighted=TRUE erzeugt wird, besitzt eine grössere Anzahl Agenten, die denen der realen Population entsprechen und deren Kombination nicht in der Stichprobe zu sehen ist (Abbildung 3.3 a, b und c). Bei den erzeugten Agenten, deren Attributenkombination aus der Stichprobe stammt und denen je eine realere Person entspricht, ist kein grosser Unterschied zu erkennen (Abbildung 3.2 a, b und c). Diese beiden Tatsachen führen dazu, dass die Durchschnittsmenge *match.intersect* der simulierten und der realen Population mit weighted=TRUE grösser ausfällt (Abbildung 3.3 d, e und f).

Fazit: Der Gewichtungsparameter weighted sollte gleich TRUE gesetzt werden, da die so erzeugte synthetische Population mehr der realen ähnelt. Dies wird durch einen höheren Wert von *match.intersect* und niedrigere Werte der Fehlerindikatoren festgestellt.

Abbildung 3.3 a, b, c, d, e und f



(a) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

(b) cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

(c) cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

(d) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

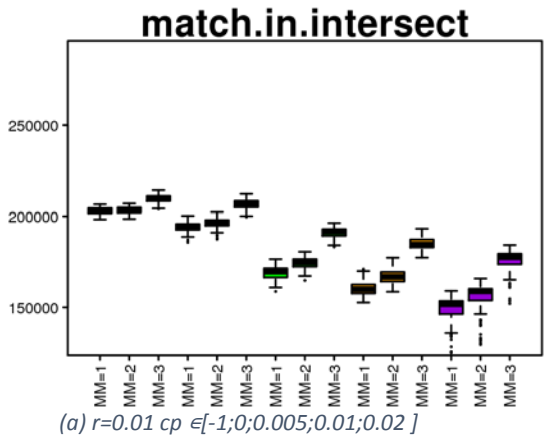
(e) cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

(f) cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

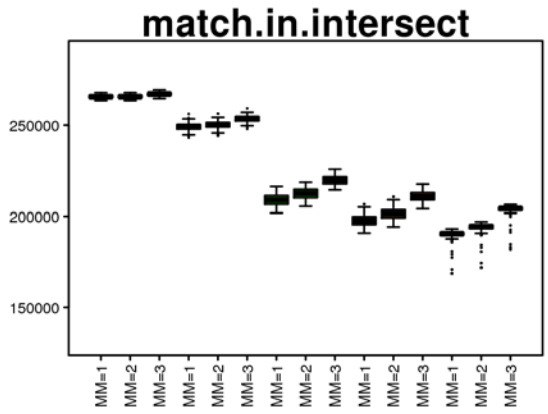
3.4 Der Filterparameter MM

Der Parameter MM wird wie der Parameter weighted mit dem Indikatorensatz indicators.1 untersucht. Auch hier wird die Abhängigkeit der Indikatoren mittels Boxplots visualisiert. Im Gegensatz zu weighted ist die Grössenordnung der Auswirkungen des Parameters MM von der Stichprobengrösse abhängig. Der Einfluss des Filterparameters ist, abgesehen von der Grössenordnung, bei jedem cp Wert analog. Die Grafiken bezüglich aller Indikatoren sind im Anhang zu finden.

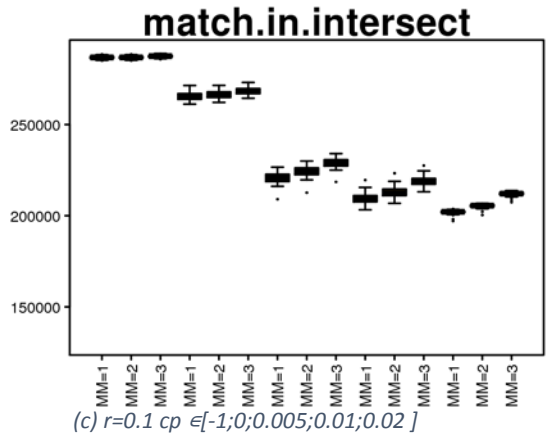
Abbildung 3.4 a, b, c, d, e, f, g, h und i



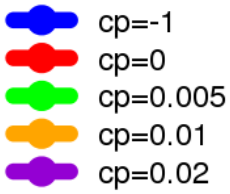
(a) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

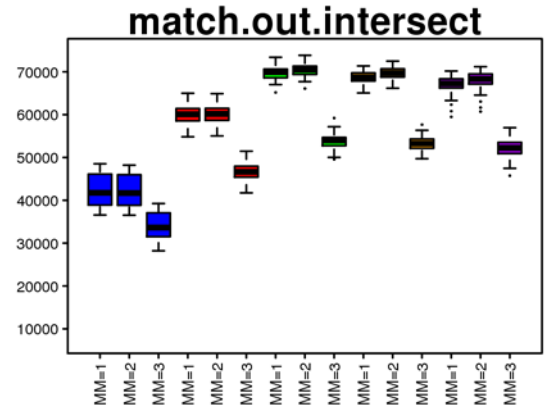


(b) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

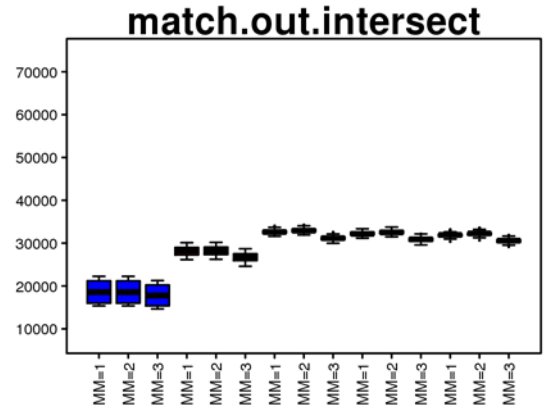


(c) r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

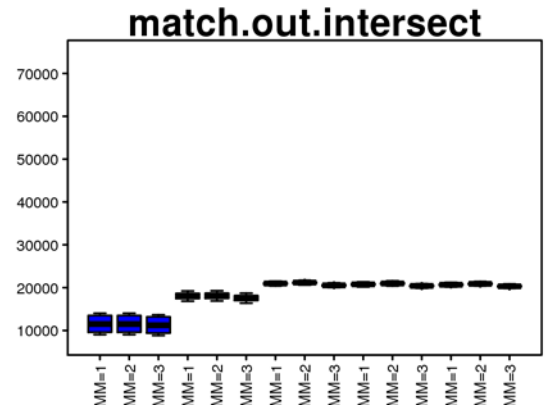


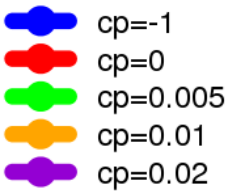


(d) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

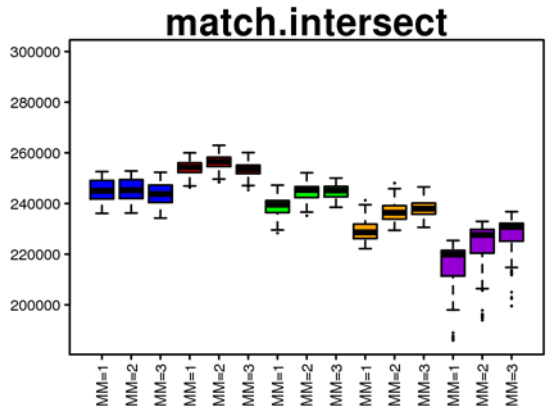


(e) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

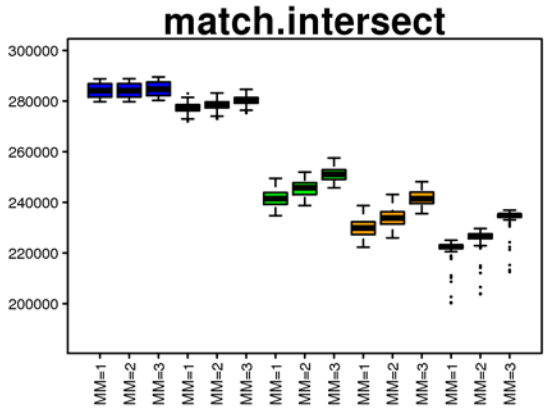




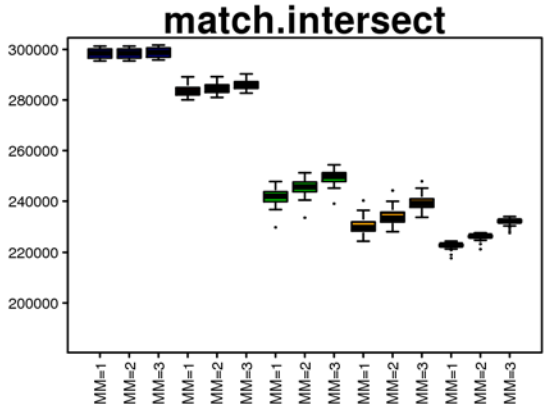
(f) r=0.1 cp ∈[-1:0:0.005:0.01:0.02]

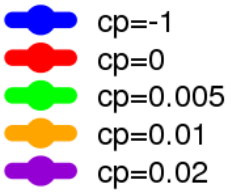


(g) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]



(h) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]





(i) r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Die Agentenmenge, deren Attributenkombination in der Stichprobe enthalten ist steigt, mit grösser werdendem MM, wobei die Ausprägung mit steigender Stichprobengrösse abnimmt (Abbildung 3.4 a, b und c). Bei der richtig erzeugten Agentenmenge, deren Kombination nicht in der Stichprobe liegt, wird das Maximum bei MM=2 erreicht und das Minimum bei MM=3(Abbildung 3.4 d, e und f). Betrachtet man die gesamte Menge erzeugter Agenten, denen eine Observation der realen Population zugewiesen werden kann so, hängt das Maximum vom cp Wert und der Stichprobengrösse ab. Bei cp grösser Null wird dieses bei MM=3 erreicht, bei r ∈ [0.05; 0.1] wird das Maximum auch für cp ∈ [-1; 0] bei MM=3 erreicht. Bei r=0.01 hingegen wird das Maximum bei cp ∈ [-1; 0] mit MM=2 erreicht (Abbildung 3.4 g, h und i).

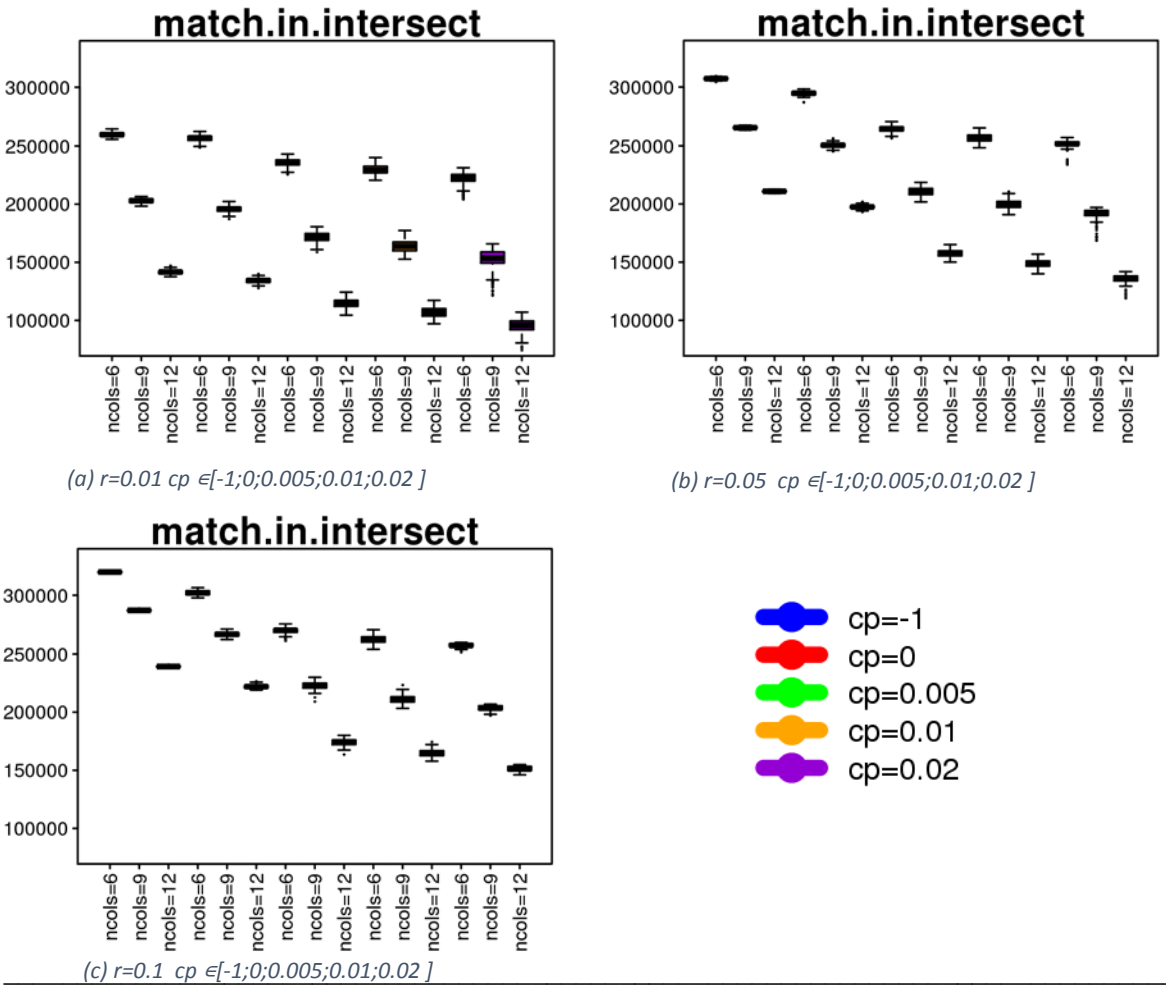
Fazit: Die Wahl des MM-Parameter ist stichprobenabhängig. Da die Rechenzeit des Algorithmus bei MM=3 um ein Vielfaches zunimmt, sollte MM=2 gewählt werden, auch wenn

in den meisten Fällen die Durchschnittsmenge zwischen der simulierten und der realen Population mit MM=3 am grössten ist.

3.5 Der Komplexitätsparameter cp

Die Untersuchung des cp-Parameters wird anhand des Indikatorensatzes indikators.2 durchgeführt. Mit diesem Parameter wird direkt in das Herz des Algorithmus eingegriffen, indem gesteuert wird, wie ausgeprägt und präzise die Regressionsbäume erstellt werden, mit denen die multivariate Wahrscheinlichkeitsverteilung erstellt wird. Die Wahl dieses Parameters beeinflusst somit am meisten die Qualität der synthetisierten Population. Um den Einfluss des cp-Parameters auf die Indikatoren match.intersect zu veranschaulichen, werden für jeden Indikator 15 Boxplots erstellt. Diese unterscheiden sich anhand der Anzahl berücksichtigter Attribute (6, 9 und 12) und des cp-Wertes (-1, 0, 0.005, 0.01, 0.02). Die Boxplots aller anderen Indikatoren sind im Anhang zu finden.

Abbildung 3.5 a, b und c



(a) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

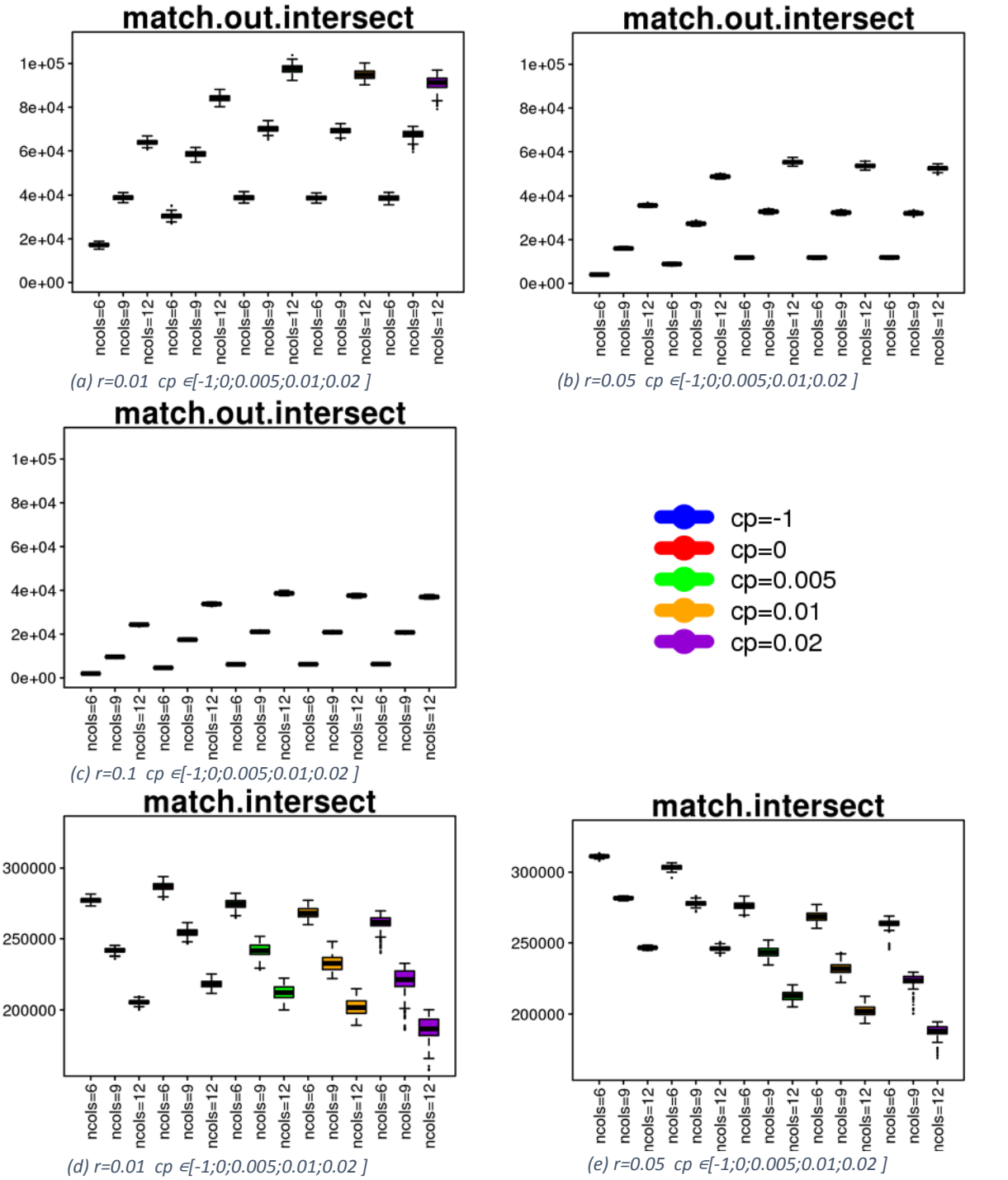
(b) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

(c) r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Abbildung 3.5 a, b und c zeigt, dass *match.in.intersect* mit den maximalen Regressionsbäumen für jedes Szenario am grössten ausfällt. Betrachtet man den Indikator *match.out.intersect* ist die entgegengesetzte Tendenz zu erkennen. Aus einer Erhöhung des cp-Wertes im Intervall [-1; 0; 0.005] resultiert für die Agenten, deren Attributenkombination nicht aus der Stichprobe stammt,

eine grössere Durchschnittsmenge zwischen der realen und der simulierten Population. Wird die Schwelle von cp=0.005 überschritten, so wird diese Menge wieder kleiner (Abbildung 3.6 a, b und c). Abbildung 3.6 d, e und f zeigt die Ausprägung der gesamten Durchschnittsmenge zwischen der realen und der simulierten Population in Funktion von cp. Bei r=0.01 ist für alle Attributenanzahlen das gleiche Muster erkennbar. Der grösste Wert für *match.intersect* wird bei cp=0 erreicht. Er wird bei einer Erhöhung von cp kleiner. Für die grösseren Stichproben hingegen ist diese Menge bei cp=-1 maximal und wird mit einer Erhöhung von cp immer kleiner.

Abbildung 3.6 a, b, c, d, e und f



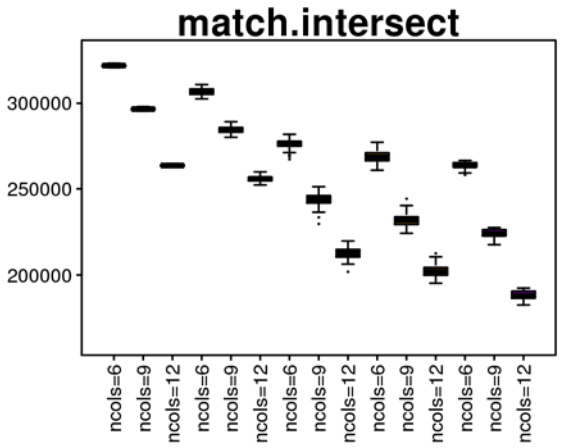
(a) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

(b) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

(c) r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

(d) r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

(e) r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



(f) r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

---

4 Einfluss der Parameter auf die Synthetische Population

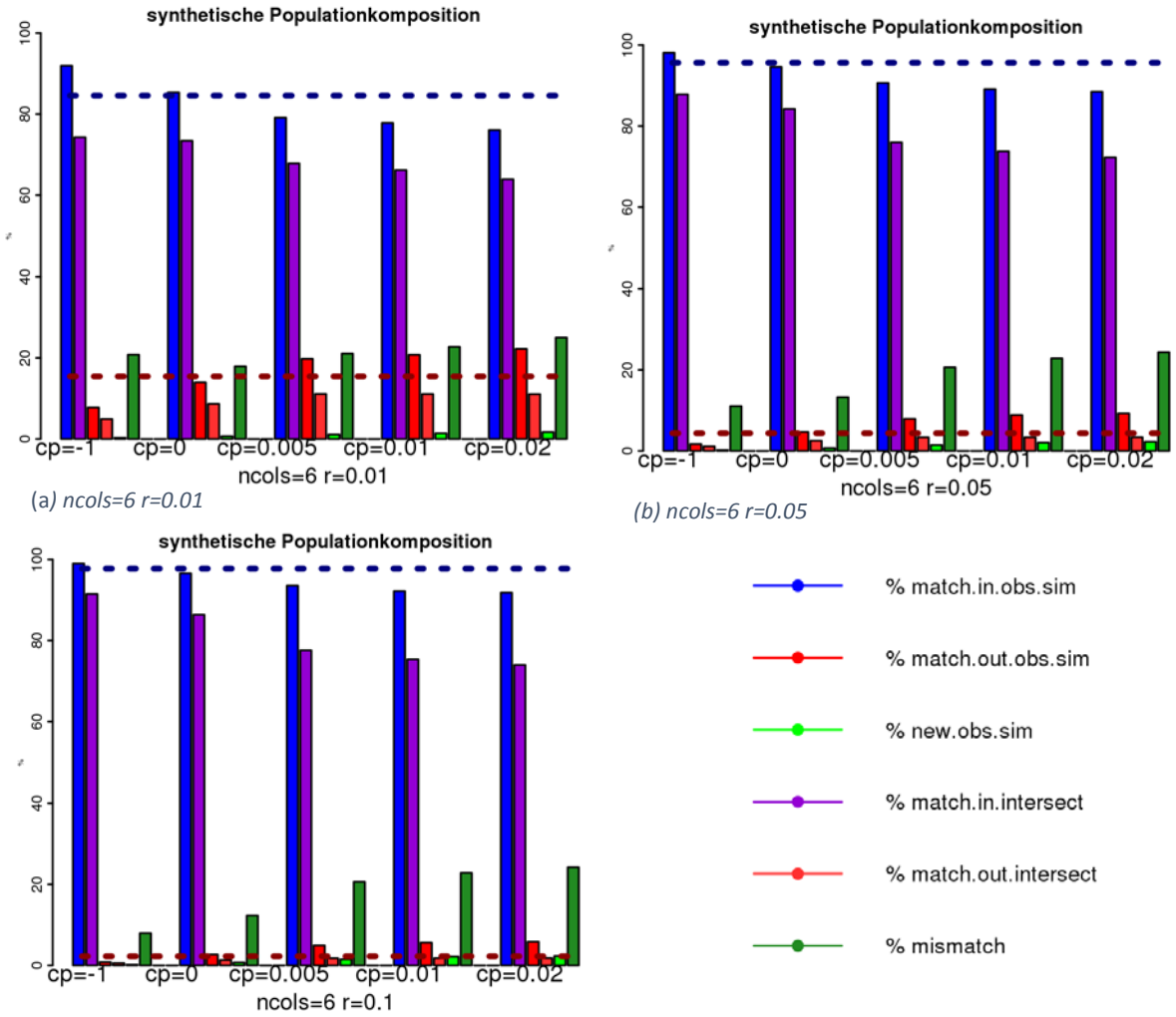
In Kapitel 3 wurden die Auswirkungen der Parameter auf die Indikatorenfamilie match.intersect untersucht, die synthetische Population in ihrer Gesamtheit aber nicht betrachtet. Aus den Untersuchungen der Parameter weighted und MM wird festgestellt, dass diese die synthetische Population nur geringfügig verändern können und deshalb nur der Feinjustierung dienen. Bei weighted war dies zu erwarten, da vom theoretischen Ansatz her keine Differenz entstehen sollte, was in R aber nicht perfekt gelingt. Der geringe Einfluss des Filterparameters könnte zwei Gründe haben. Erstens wurde dieser nur mit dem Datensatz indicators.1 Untersucht, bei 9 Attributen. Zweitens könnte dies an der CART-Methodologie liegen, die den Kombinationen die durch MM ausgeschlossen werden, schon vorweg eine sehr geringe Auftretenswahrscheinlichkeit zuweist. Daher fällt die Differenz der synthetisierten Populationen mit verschiedenen MM-Werten bescheiden aus. Das wichtige „tuning“ des Algorithmus wird durch den Komplexitäts Parameter vollzogen, welcher die Qualität und die Zusammensetzung der synthetischen Population bestimmt.

4.1 Populationskomposition

Es wird nun die gesamte synthetische Population untersucht, die man in gleichen Szenarien mit verschiedenen cp-Werten erhält. Die nachfolgenden Abbildungen zeigen die Zusammensetzung der synthetischen Population. Diese besteht aus den drei Agententypen *new.obs.sim*, *match.in.obs.sim* und *match.out.obs.sim*, also den Agenten deren Attributenkombination keiner Person in der realen Population entspricht, deren Attributenkombinationen gleich der von Personen aus der Stichprobe ist und deren Attributenkombination in der realen Population zu finden ist, aber nicht in der Stichprobe. Um diese Mengen zu bewerten, werden in jeder Grafik die Indikatoren *match.in.intersect*, *match.out.intersect* und *mismatch* hinzugefügt. *Mismatch* entspricht der Anzahl Personen, die nicht richtig auf einen synthetischen Agenten abgebildet werden. Alle Werte werden normiert, sodass sie den Mengenanteil in Bezug auf die gesamte synthetische Population in Prozent wiederspiegeln. Die gesamte reale Population wird in zwei Untermengen gespalten, eine mit den Personen deren Attributenkombination in der Stichprobe zu finden ist (blaue gestrichelte Linie) und eine wo dies nicht der Fall ist (rote gestrichelte Linie). Diese Werte werden durch

den Durschschnitt über die 100 Stichproben ermittelt. Der Anteil dieser Mengen wird in Prozent angegeben und durch die gestrichelten Linien in den Grafiken repräsentiert. Die untersuchten Szenarien sind die der Kombinationen MM=2, weighted=FALSE, ncols ∈ [6; 9; 12], r ∈ [0.01; 0.05; 0.1] und cp ∈ [-1; 0; 0.005, 0.01; 0.02].

Abbildungen 4.1 a, b und c



(a) *ncols=6 r=0.01*

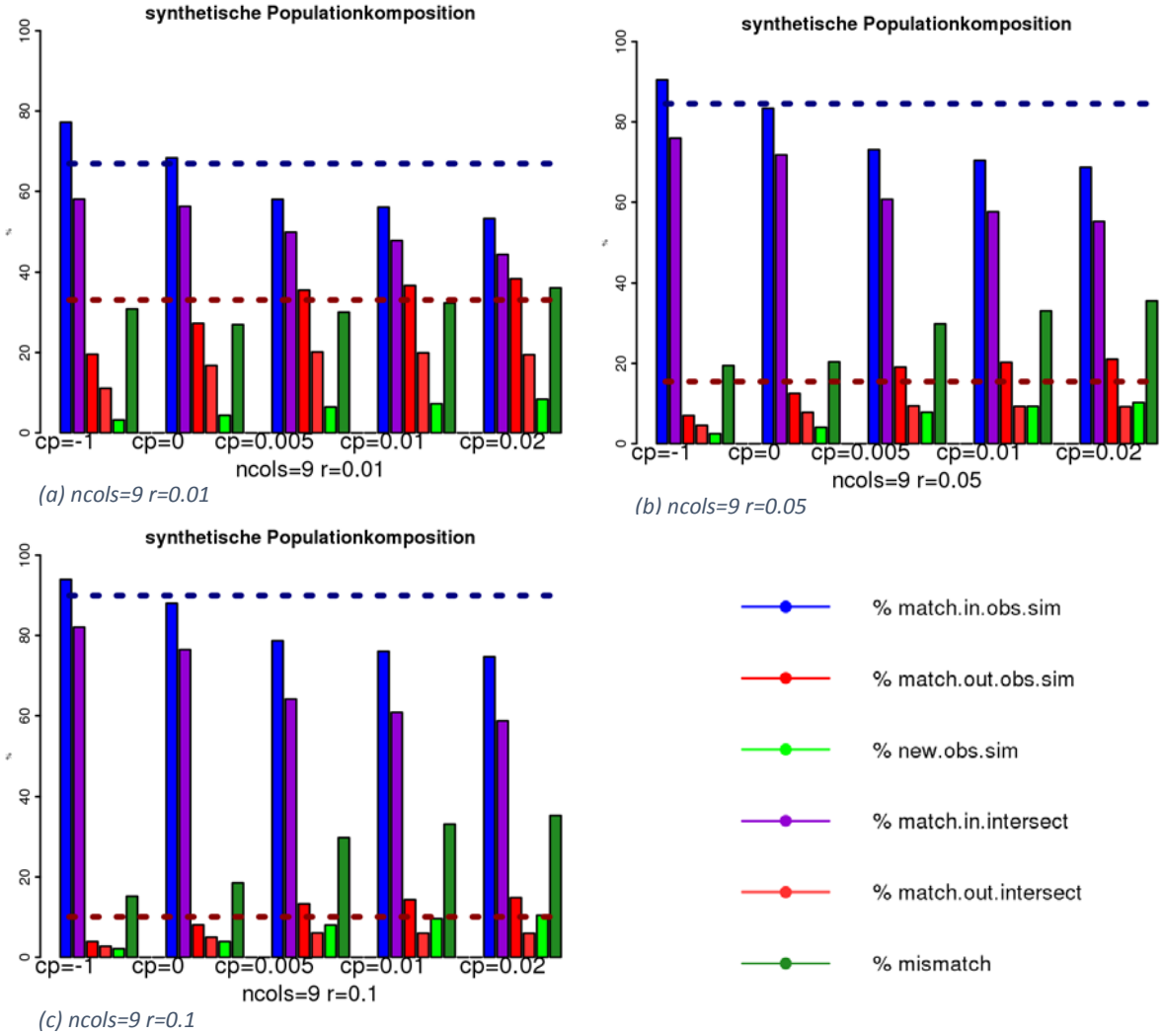
(b) ncols=6 r=0.05

(c) ncols=6 r=0.1

Abbildung 4.1 zeigt die Zusammensetzung der synthetischen Population, die durch sechs Attribute beschrieben wird. Da es sich um eine geringe Attributenanzahl handelt, ist offensichtlich, dass die Stichprobe die meisten Kombinationen der realen Population enthält, was durch eine niedrige Lage der roten Linien belegt wird. Man kann erkennen, dass die qualitativ hochwertigere Population in Abhängigkeit der Stichprobengrösse, anhand verschiedener cp Werte erzeugt wird. Bei einer kleinen Stichprobe (Abbildung 4.1a) wird der fehlerhafte Anteil der Population mit cp=0 minimiert, während dies bei grösseren Stichproben mit cp=-1 erreicht wird (Abbildung 4.1 b und c). Dies kann anhand der Modell-Erstellung verstanden werden. Bei cp=-1 wird der Algorithmus in stärkeren Mass an die Daten der Stichprobe angepasst, was bei den Stichprobengrössen r ∈ [0.05; 0.1] zu besseren Ergebnissen führt, da die Korrelationsstruktur der Attribute der Stichprobe sich nur in geringer Weise von der der realen Population unterscheidet. Bei r=0.01 ist dies nicht mehr der Fall und daher ist es notwendig, Agenten mit Attributenkombinationen zu synthetisieren, die sich von denen der

Stichprobe unterscheiden. Dies wird durch grössere cp Werte erreicht, wobei cp=0 für dieses Szenario die beste synthetische Population liefert.

Abbildungen 4.2 a, b und c



(a) ncols=9 r=0.01

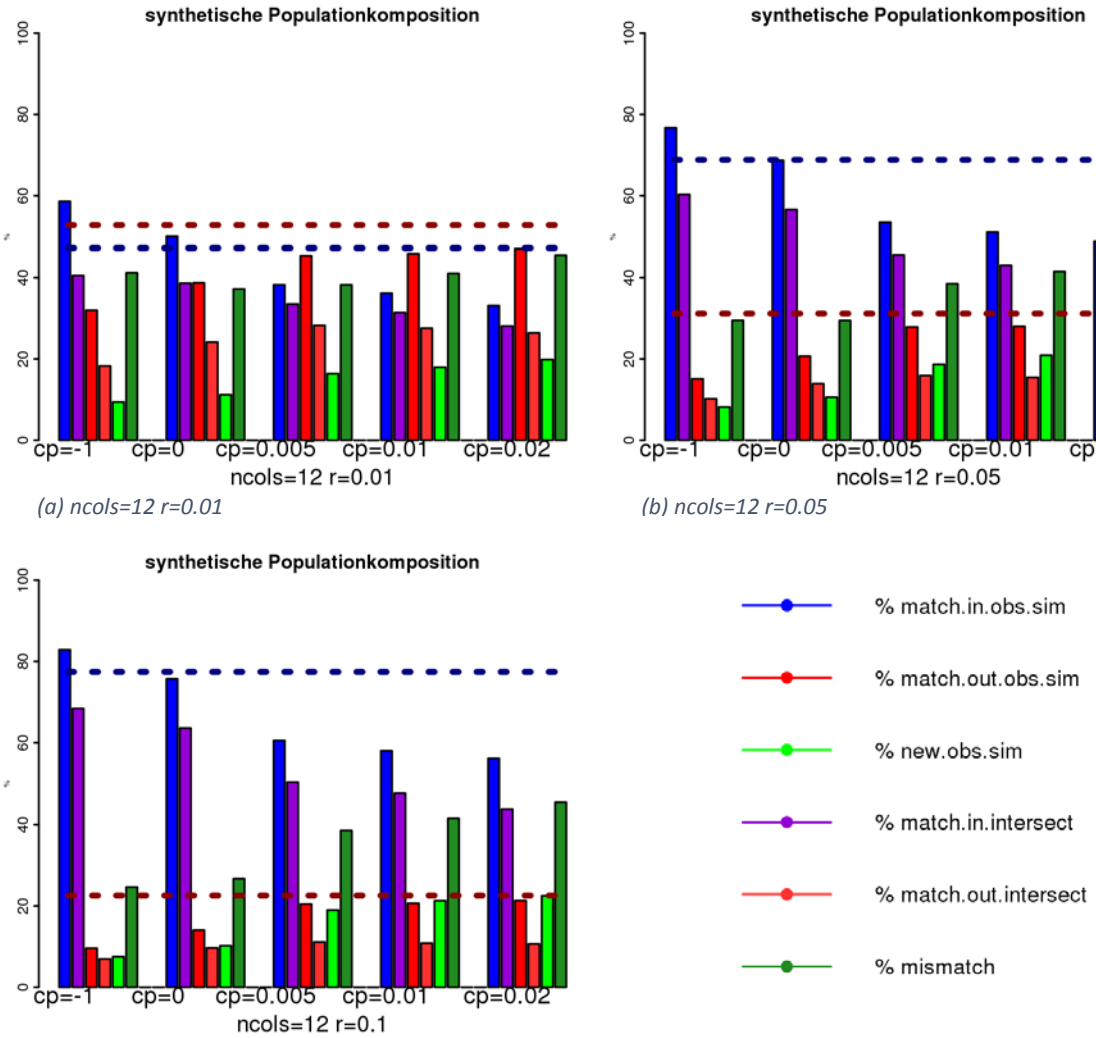
(b) ncols=9 r=0.05

(c) ncols=9 r=0.1

Abbildung 4.2 zeigt die synthetische Populationskomposition in Szenarien mit neun Attributen. Im Vergleich zum vorherigen Szenario, decken die Personen der Stichprobe einen geringeren Teil aller Kombinationen der realen Population ab, was durch eine niedriger liegende, blaue Linie zu erkennen ist. Dies ist die Folge der Erhöhung der Attributenanzahl, die eine Vergrösserung der möglichen Kombinationen mit sich bringt, wobei die Stichprobengrösse gleich bleibt. Trotzdem bleibt die Menge, die durch eine Expansion der Stichprobe erhalten werden kann, dominierend in der realen Population. Bei r ∈ [0.01; 0.05] wird mit cp=-1 die Observationsmenge des Typs match.in zu gross und ungenau synthetisiert, was in Abbildung 4.2 a und b zu sehen ist. Cp=0 scheint in diesem Szenario die beste Wahl zu sein, da die Region der match.in Agenten in der richtigeren Menge erzeugt wird, und mit einer besseren Ausbeute, was durch die geringere Differenz der blauen und der Violetten Histogramme im Vergleich zu cp=-1 zu sehen ist. Bei r=0.1 führt ein höherer Anpassungsgrad des Algorithmus an die Daten, mit denen er kalibriert wird, zur besten Population und deshalb ist cp=-1 die bessere Wahl (Abbildung 4.2 c). Cp ∈ [0.005; 0.01; 0.02] bewirken zwar eine Vergrösserung von

match.out.intersect, aber da sie eine grössere Verkleinerung der *match.in.intersect*-Menge zur Folge haben, müssen sie verworfen werden.

Abbildungen 4.3 a,b und c



(a) ncols=12 r=0.01

(b) ncols=12 r=0.05

(c) ncols=12 r=0.1

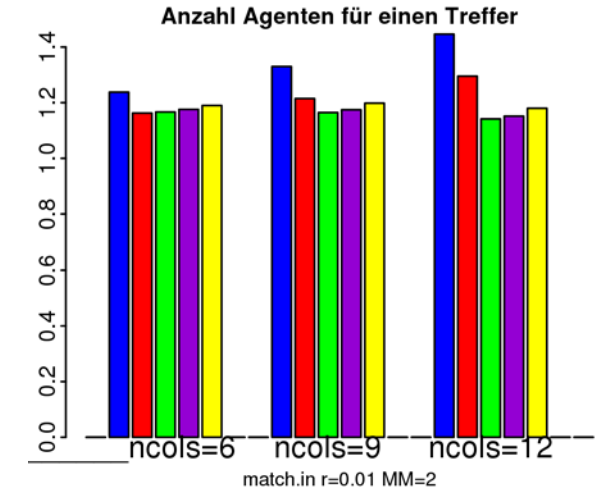
Die Szenarien von Abbildung 4.3 unterscheiden sich deutlich von den vorherigen. Der „in sample"-Teil ist bei r=0.01 sogar kleiner als der „out of sample"-Teil (Abbildung 4.3 a). Dies hat zu Folge, dass die Menge match.out einen gleichwertigen Beitrag zur Qualität der synthetischen Population liefert wie der match.in Anteil. Deshalb muss der Algorithmus so eingestellt werden, dass er mehr Agenten synthetisiert, deren Attributenkombination verschieden von denen ist, die in der Stichprobe enthalten sind. Dies wird durch die Erzeugung kleinerer Bäume erreicht, die entstehen, wenn ein grösserer cp-Wert gewählt wird. Cp=0 liefert dabei für $r \in [0.01; 0.05]$ die besten Ergebnisse (Abbildung 4.3 a und b). Bei r=0.1 ist die Stichprobe wieder aussagekräftiger und somit lohnt es sich, cp=-1 zu wählen, da ein besser an die Stichprobendaten angepasstes Modell dieselben besser expandiert. Da die Agenten des Typs match.in den grössten Teil der realen Population repräsentieren, wird somit eine bessere synthetische Population erzeugt (Abbildung 4.3 c).

4.2 Trefferquote

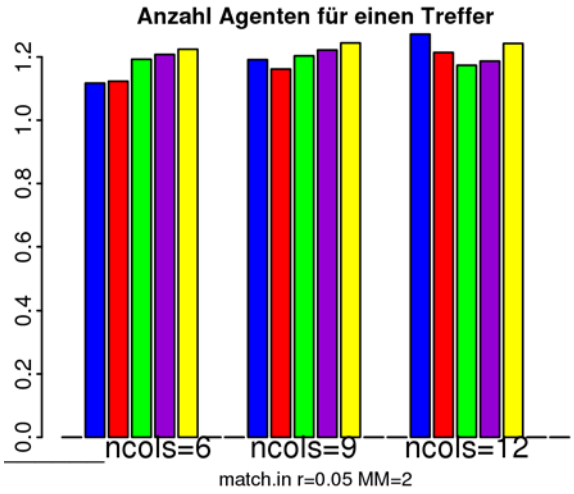
In diesem Absatz wird das Verhältnis der Anzahl erzeugten Observationen, deren Attributenkombination in der realen Population existiert, mit der die es in der realen Population gibt untersucht. Dies wird gemacht in dem die Trefferquote analysiert wird. Die Frage ist dabei, wie viele Agenten des Typ match erzeugt werden müssen, um einen Agenten des Typs match.intersect zu erzeugen.

$$\text{Trefferquote}=\frac{match.in.obs.sim}{match.in.intersect} \text{ bzw. } \frac{match.out.obs.sim}{match.out.intersect}$$

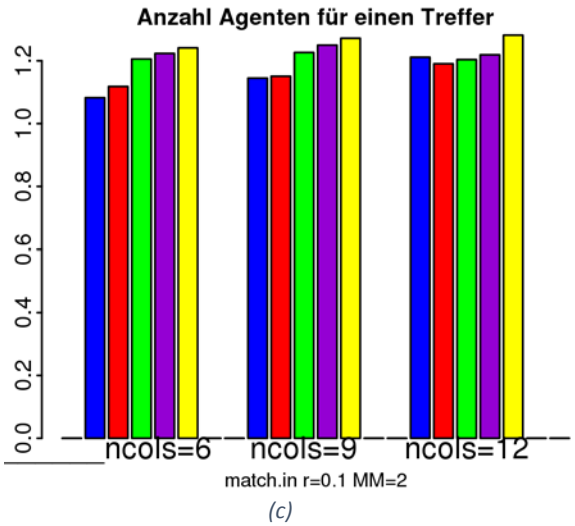
Abbildung 4.4 a, b, c, d, e und f



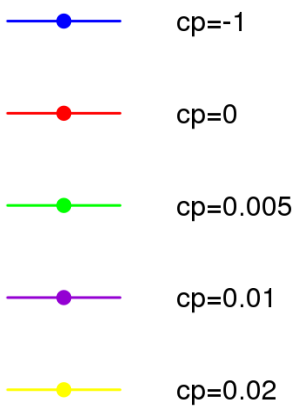
(a)



(b)



(c)



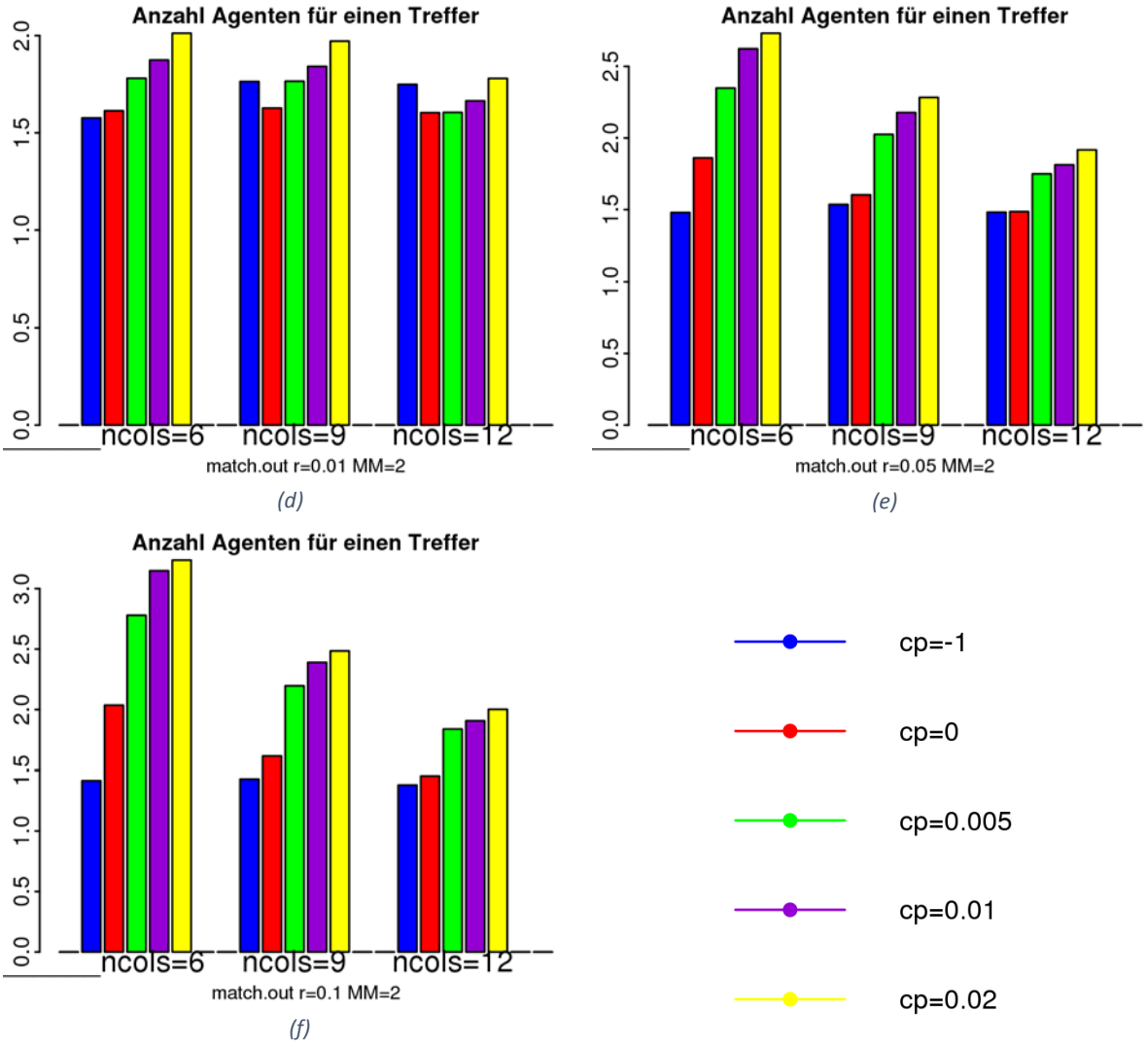


Abbildung 4.4 a,b und c zeigt die Trefferquote für die synthetischen Agenten des Typs match.in mit 6, 9 und 12 berücksichtigten Attributen, r ∈ [0.01; 0.05; 0.1] und MM=2, für die verschiedenen cps. Es ist deutlich zu erkennen, dass die Trefferquote von dem Szenario abhängt, also von r und ncols. Für ein Szenario mit vielen relevanten Attributen und kleinen Stichproben, führt cp=-1 zu einer Überrepräsentierung von *match.in.obs.sim* (Abbildungen 4.1a, 4.2a, 4.3a und b), was sich negativ auf die Trefferquote auswirkt (Abbildung 4.4a). Bei kleinen Attributenkombinationslängen und grösseren Stichproben hingegen geschieht das Gegenteil, da die Stichprobe die Kombinationsmenge der realen Population optimal abdeckt und so der Observationstyp match.in der am weitesten verbreitete in der realen Population ist. Dies wirkt sich positiv auf die Trefferquote aus (Abbildung 4.4b und c). Grössere cp Werte hingegen führen bei kleinen Stichproben und grösserer Attributenanzahl zu einer besseren Trefferquote (Abbildung 4.4a und b) da sie die Teilmenge match.in nicht in übertriebener Grösse synthetisieren. Abbildung 4.4 zeigt, dass die besten Trefferquoten immer mit cp ∈ [-1; 0; 0.005] erreicht werden. Die Erkenntnisse über die Trefferquote des Typs match.in können folgendermassen zusammengefasst werden:

Cp=-1: gut geeignet für grosse Stichproben und kleine Attributenanzahl

Cp=0: weist seine Stärken bei kleinen Stichproben mit kleiner Attributenanzahl und für mittelgrosse Stichproben mit hoher Attributenanzahl auf.

Cp=0.005: ist die beste Wahl bei kleinen Stichproben und langen Attributenkombinationen

Betrachtet man die Trefferquote der Observationen des Typs match.out. (Abbildung 4.4 d, e und f) ist zu erkennen, dass die besten Ergebnisse wieder ausschliesslich mit cp Werten ∈ [-1; 0] erhalten werden. Dabei können folgende Aussagen gemacht werden:

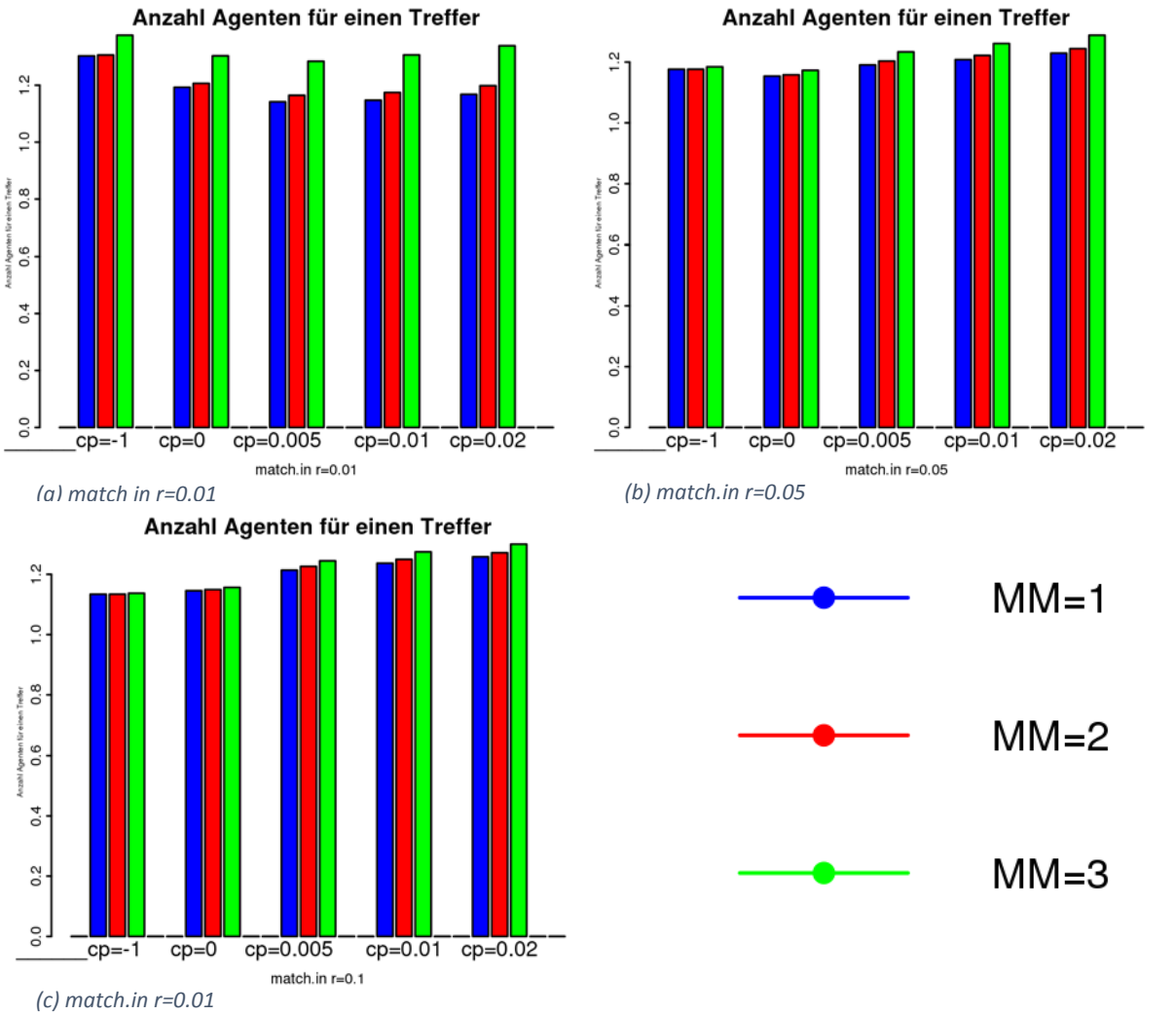
Cp=-1: besitzt die beste Trefferquote bei kleinen Stichproben mit kleinen Atrtributenkombinationslängen und bei mittelgrossen bis grossen Stichproben für alle Attributenkombinationen

Cp=0: bestens geeignet für kleine Stichproben und grosse Attributenkombinationslängen

Was noch zu sehen ist, ist die Differenz der Trefferquote zwischen match.in und match.out. Letztere fällt deutlich geringer aus.

Im Folgenden wird der Einfluss des Filterparameters MM auf die Trefferquote untersucht (Abbildung 4.5 und 4.6). Dabei werden die Ergebnisse mit MM=1 in Blau, die von MM=2 in Rot und die von MM=3 in Grün eingetragen. Die Untersuchung erfolgt mit den Mittelwerten der Indikatoren indicators.1, also mit 9 Attributen.

Abbildung 4.5 a, b und c



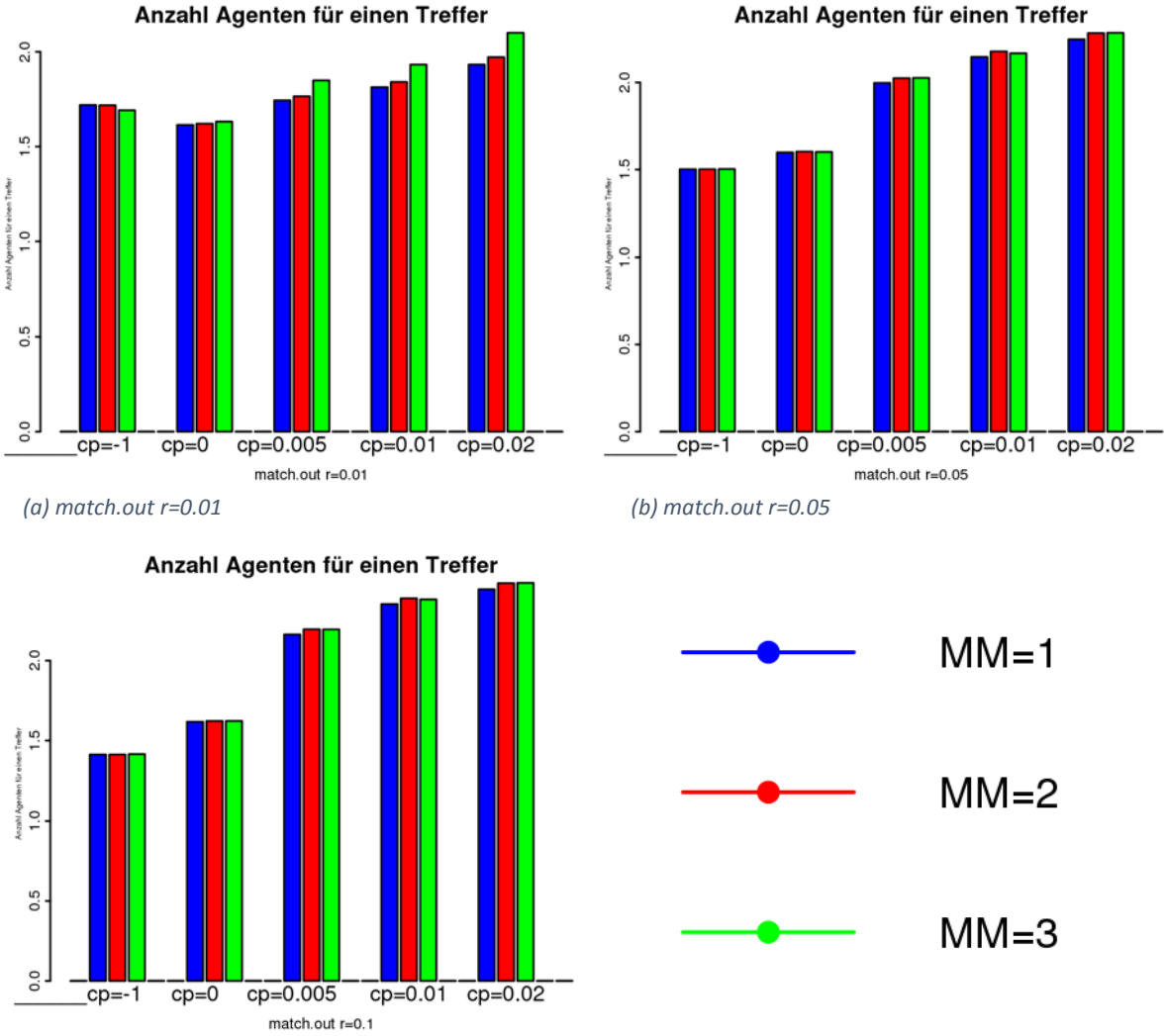
(a) match in r=0.01

(b) match.in r=0.05

(c) match.in r=0.01

Abbildung 4.5 zeigt die Auswirkung des Filterparameters auf die Trefferquote des Typs match.in. Man sieht deutlich, dass eine Erhöhung von MM sich immer negativ auf die Trefferquote des Typs match.in auswirkt.

Abbildung 4.6 a, b und c



(a) match.out r=0.01

(b) match.out r=0.05

(c) match.out r=0.1

Abbildung 4.6 zeigt den Einfluss von MM auf die Trefferquote des Typs match.out. Auch hier folgt einer Erhöhung von MM eine Verschlechterung der Trefferquote. Dies ist aber nicht negativ zu bewerten da ein grösseres MM eine geringere Anzahl von Agenten des Typs new zur Folge hat. Eine Verkleinerung dieser Menge führt zu einer Vergrösserung der Mengen match.in und match.out, was sich negativ auf die Trefferquote auswirkt.

5 Parameterwahl

Die vorab durchgeführten Analysen haben gezeigt, dass die beste Parameterwahl für jedes Szenario anders ausfällt. Die Wahl des Parameters cp ist nicht ganz unproblematisch. Dieser sollte aber im Intervall [-1, 0] liegen, da bei grösseren cp Werten der match.in Teil der synthetischen Population zu gering ausfällt. Da dieser bei den untersuchten Daten den grössten Teil der realen Population ausmacht, entsteht ein erheblicher Qualitätsverlust. Die Wahl zwischen cp=-1 oder 0 hängt von der Stichprobengrösse und der Anzahl berücksichtigter Attribute ab. Bei kleinen Stichproben ($r \in [0.01, 0.02]$) und grösseren Attributenkombinationslängen wird mit cp=0 die bessere synthetische Population erzeugt. Cp=-1 bewirkt in diesem Szenario eine Überrepräsentierung des „in sample“-Anteils der Population, was sich negativ auf die Erzeugung von Agenten des Typs match.out auswirkt. Wird hingegen der Kombinationsgehalt der Stichprobe im Vergleich zur realen Population grösser, was bei geringen Attributenanzahlen und grossen Stichproben der Fall ist, wird cp=-1 die bessere Wahl, da der „in sample“-Anteil der Population korrekter synthetisiert wird. Da mit einem höheren Informationsgehalt der Stichprobe fast die ganze reale Population erhalten werden kann, führt eine bessere Behandlung des match.in Typs zu einer insgesamt besseren synthetischen Population.

6 Schlussbemerkungen

Wegen der geringen zur Verfügung stehenden Zeit konnte das Verfahren nicht in allen Szenarien getestet werden und vollständig ausgewertet werden. Es muss berücksichtigt werden, dass der Algorithmus sich noch in der Entwicklungs-Phase befindet und daher seine maximale Leistungsfähigkeit noch nicht erreicht hat. Der nächste Schritt wird die Änderung der Reihenfolge der behandelten Attribute bei der Erstellung der multivariaten Wahrscheinlichkeitsverteilung betreffen. Dies müsste die Qualität der synthetisierten Population verbessern.

7 Danksagung

Ich möchte mich bei Herrn Professor Dr. K. W. Axhausen bedanken, der mir die Teilnahme an einer echten wissenschaftlichen Recherche ermöglicht hat. Besonderer Dank gilt auch Herrn Müller für das Bereitstellen der Simulationsumgebung, die Durchführung der Simulationsläufe und die Berechnung der Indikatoren und nicht zuletzt auch für seine grosse Geduld und Hilfsbereitschaft, ohne die diese Arbeit nicht zu Stande gekommen wäre.

8 Literaturangaben:

Auld, J. and A. K. Mohammadian (2010) Efficient methodology for generating synthetic populations with multiple control levels. *Transportation Research Record: Journal of the Transportation Research Board* p. 138-147.

Beckman, R. J., K. A. Baggerly and M. D. McKay (1996) Creating synthetic baseline populations. *Transportation Research Part A: Policy and Practice* p. 415-429.

Bishop, Y. M. M., S. E. Fienberg and P. W. Holland (1975) Discrete multivariate analysis: theory and practice. *MIT Press. Cambridge, MA.*

Breiman, L., J. H. Friedman, R. A. Olshen and C. J. Stone (1984) Classification and regression trees. *Wadsworth & Brooks. Monterey, CA.*

Cho, S., T. Bellemans, L. Creemers, L. Knapen, D. Janssens and G. Wets (2013) Synthetic Population Techniques in Activity-Based Research. *Data on Science and Simulation in Transportation*, p 48-60.

Deming, W. E. and F. F. Stephan, (1940) On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics* 11.4 p. 427-444.

Farooq, B., M. Bierlaire, R. Hurtubia, and G. Flötteröd (2013) Simulation based population synthesis. *Transportation Research Part B: Methodological*, *58* p. 243-263.

Geman, S. and D. Geman (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6) p. 721-741.

Guo, J. Y. and C. R. Bhat (2007) Population synthesis for microsimulating travel behavior. *Transportation Research Record: Journal of the Transportation Research Board 2014.1* p. *92-101.*

Hettinger, T. (2007) Anpassung von Aktivitätenketten mittels wiederholter proportionaler Anpassung. *Semesterarbeit für den MSc Angewandte Mathematik*. Institut für Verkehrsplanung und Transportsysteme (IVT), ETH Zürich, Zürich.

Kao, S. C., H. K. Kim, C. Liu, X Cui, and B.L. Bhaduri (2012) Dependence-Preserving Approach to Synthesizing Household Characteristics. *Transportation Research Record: Journal of the Transportation Research Board* p. 192-200.

Müller, K. and G. Flötteröd (2013), Population synthesis with regression trees. *Unveröffentlicht.*

Müller, K. and K.W. Axhausen (2011) Generating a synthetic population for Switzerland. In *ERSA conference papers, European Regional Science Association.*

Rich, J. and I. Mulalic (2012) Generating synthetic baseline populations from register data. *Transportation Research Part A* 46 p. 467-479.

Ryan, J., H. Maoh and P. Kanaroglou (2009) Population synthesis: Comparing the major techniques using a small, complete population of firms. *Geographical analysis* p. 181-203.

Stephan, F. F. (1942) An iterative method of adjusting sample frequency tables when expected marginal totals are known. *The Annals of Mathematical Statistics*, *13*(2) p. 166-178.

Voas, D. and P. Williamson (2001). Evaluating goodness-of-fit measures for synthetic microdata. *Geographical and Environmental Modelling* p. 177-200.

Wongchavalidkul, N. and P. Mongkut (2009) Estimating synthetic baseline population distribution when only partial marginal information is available. *Proceedings of the Eastern Asia Society for Transportation Studies*. Vol. 7.

Ye, X., K. Konduri, R. M. Pendyala, B. Sana and P. Waddell (2009) A methodology to match distributions of both household and person attributes in the generation of synthetic populations. *88th Annual Meeting of the Transportation Research Board*.

6 Anhang:

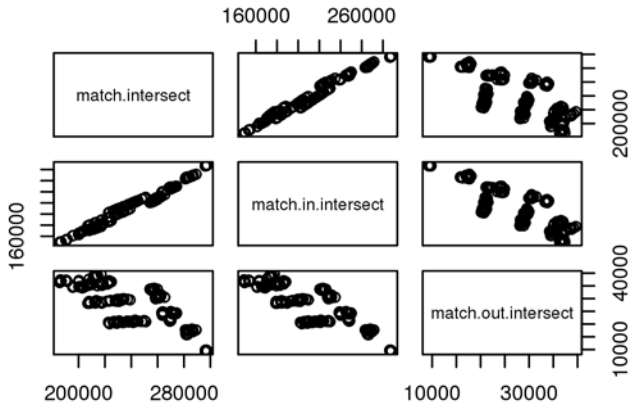


Abbildung 6.1 a ncols ∈ [9, 10 ,11, 12] r=0.1

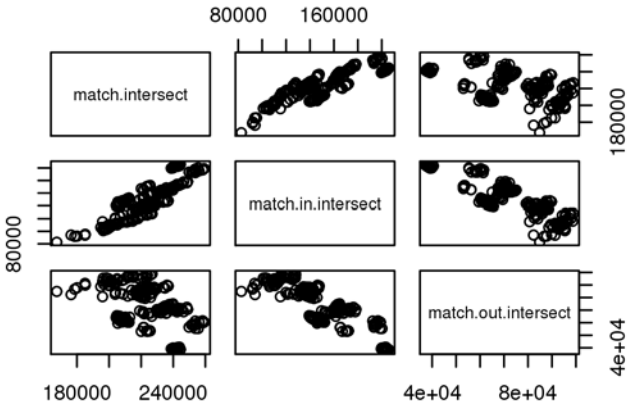


Abbildung 6.1 b ncols ∈ [9,10 ,11, 12] r=0.01

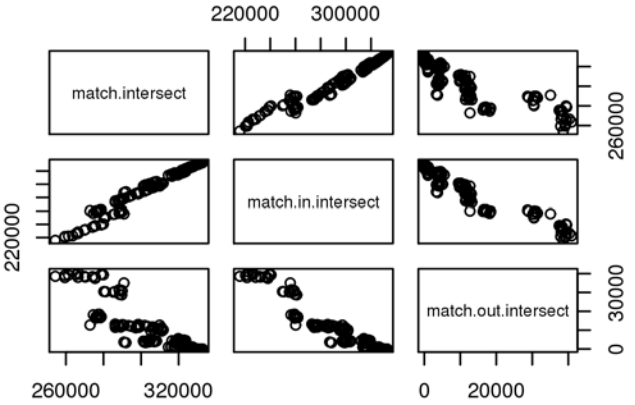


Abbildung 6.1 c ncols ∈ [3, 4, 5, 6] r=0.1

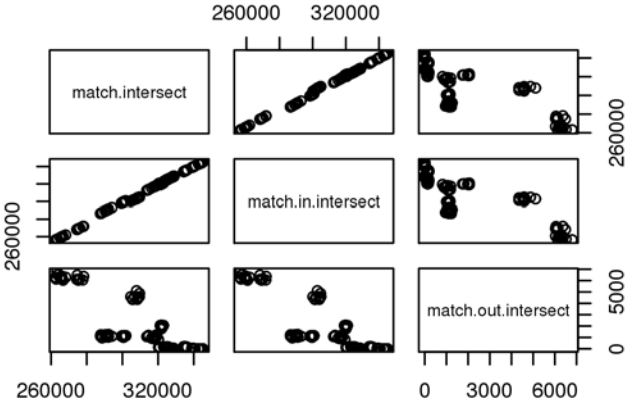


Abbildung 6.1 d ncols ∈ [3, 4, 5, 6] r=0.01

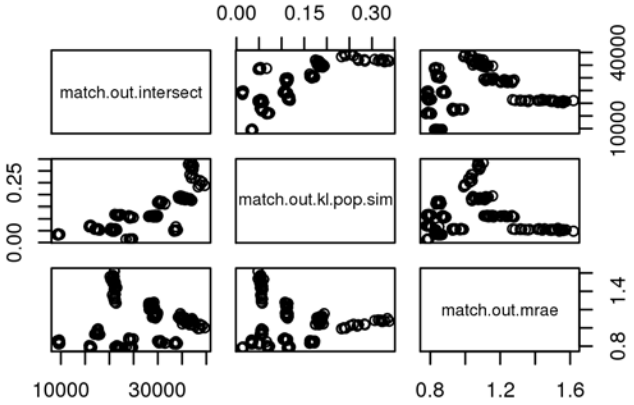


Abbildung 6.2 a ncols ∈ [9,10 ,11, 12] r=0.1

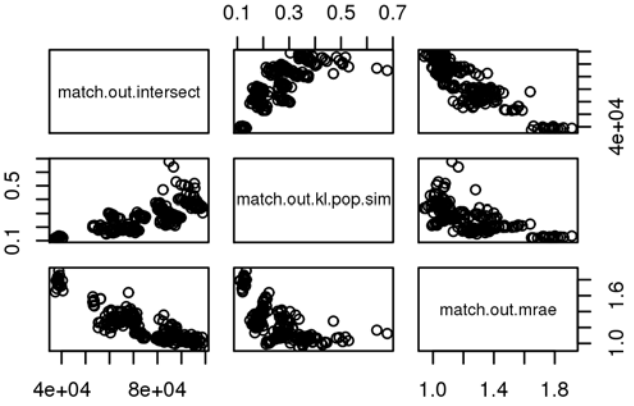


Abbildung 6.2 b ncols ∈ [9,10 ,11, 12] r=0.01

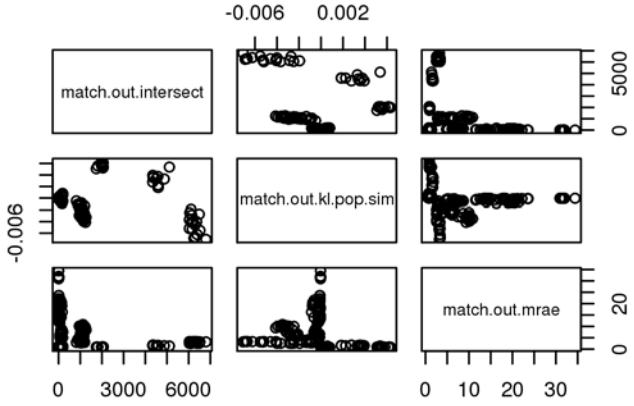


Abbildung 6.2 c ncols ∈ [3,4 ,5, 6] r=0.1

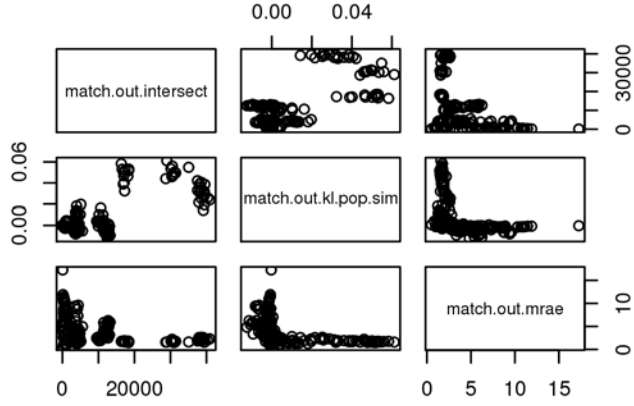


Abbildung 6.2 d ncols ∈ [3,4 ,5, 6] r=0.01

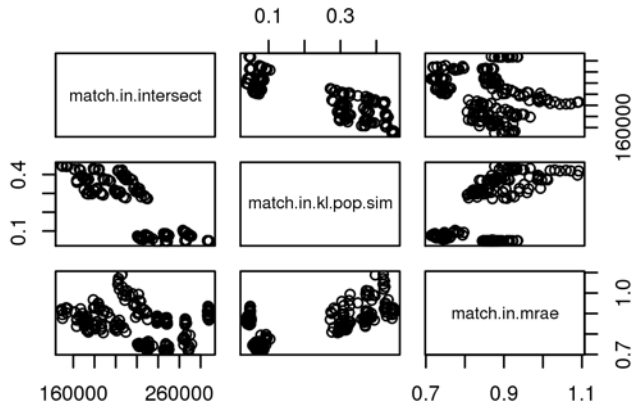


Abbildung 6.3 a ncols ∈ [9,10 ,11, 12] r=0.1

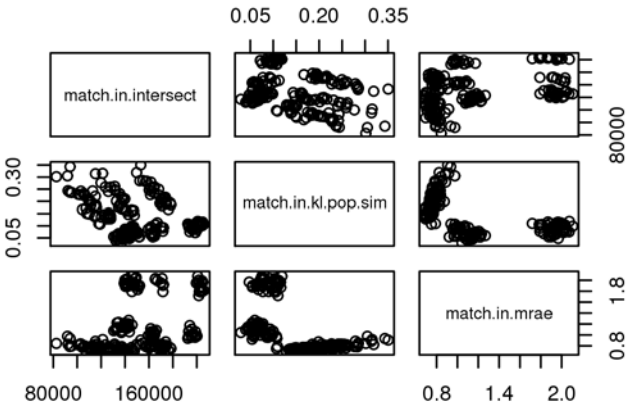


Abbildung 6.3 b ncols ∈ [9,10 ,11, 12] r=0.01

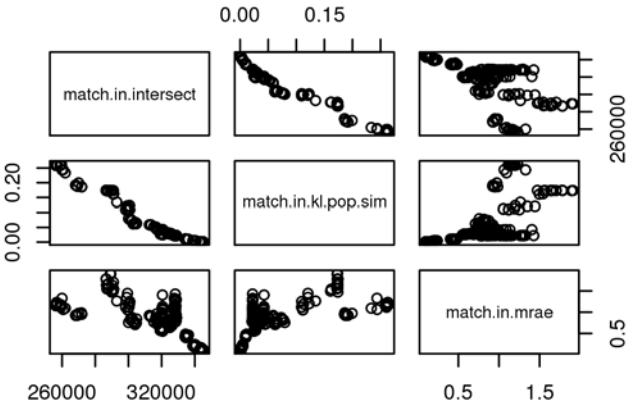


Abbildung 6.3 c ncols ∈ [3,4 ,5, 6] r=0.1

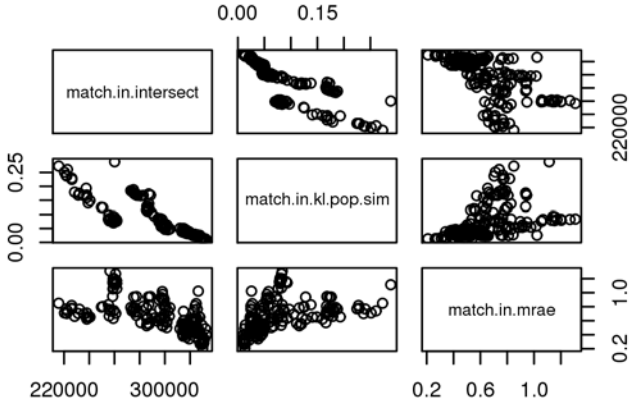


Abbildung 6.3 d ncols ∈ [3,4 ,5, 6] r=0.01

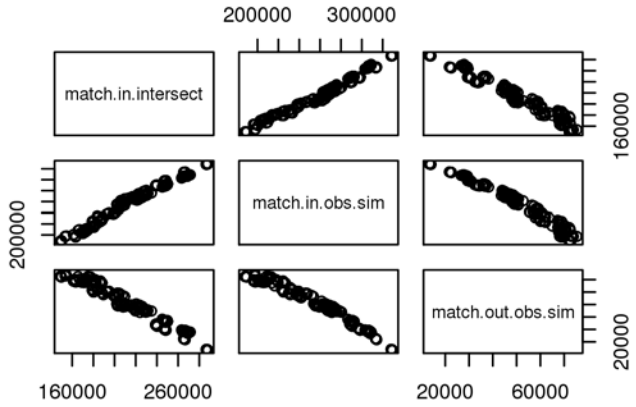


Abbildung 6.4 a ncols ∈ [9,10 ,11, 12] r=0.1

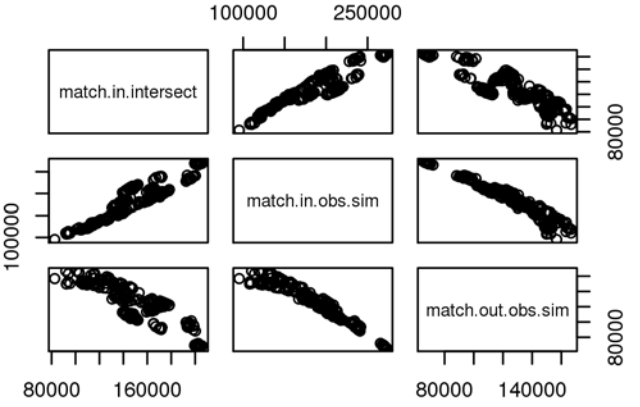


Abbildung 6.4 b ncols ∈ [9,10 ,11, 12] r=0.01

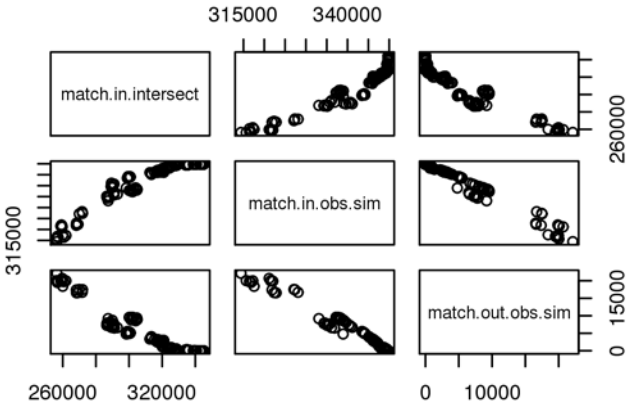


Abbildung 6.4 c ncols ∈ [3,4 ,5, 6] r=0.1

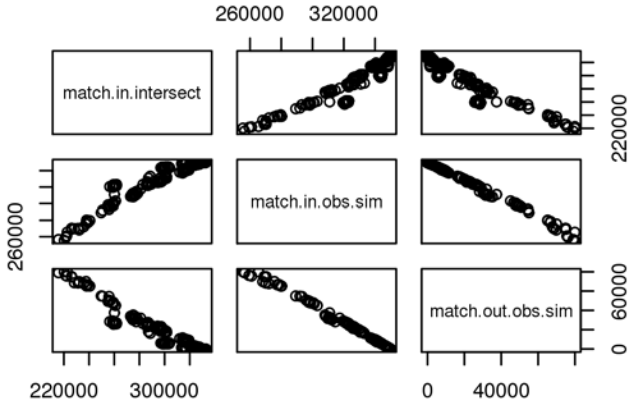


Abbildung 6.4 d ncols ∈ [3, 4 , 5, 6] r=0.01

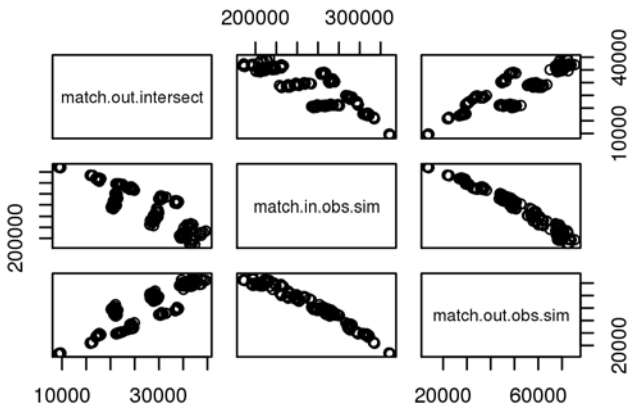


Abbildung 6.5 a ncols ∈ [9,10 ,11, 12] r=0.1

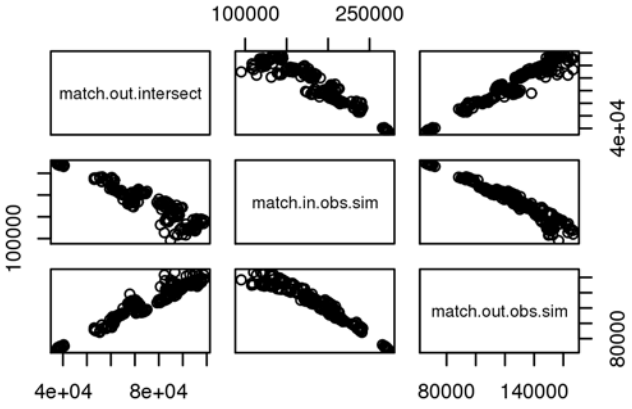


Abbildung 6.5 b ncols ∈ [9,10 ,11, 12] r=0.01

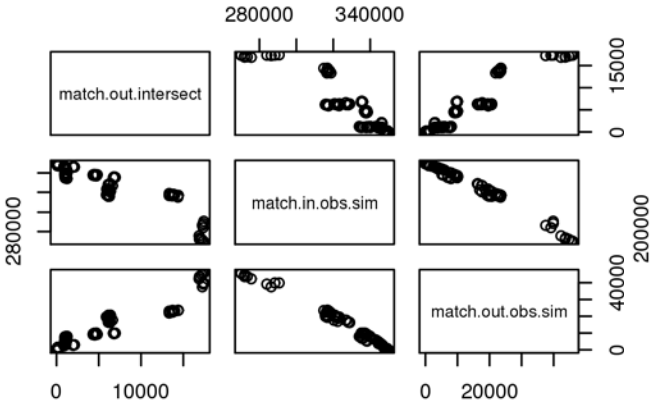


Abbildung 6.5 c ncols ∈ [3,4 ,5, 6] r=0.1

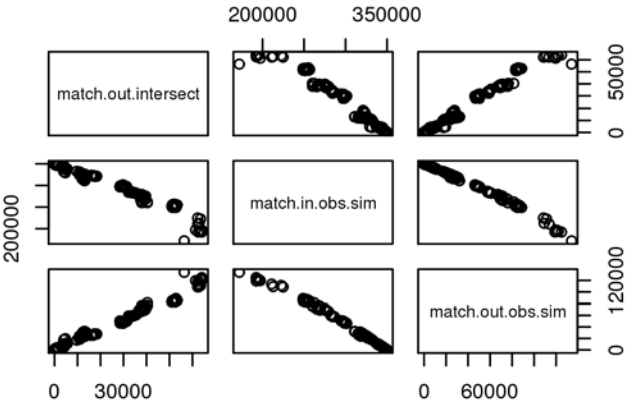


Abbildung 6.5 d ncols ∈ [3,4 ,5, 6] r=0.01

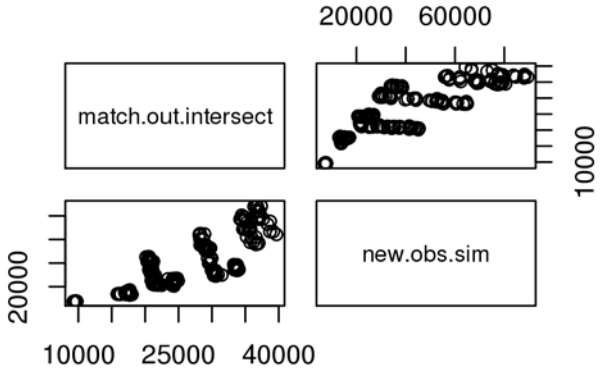


Abbildung 6.6 a ncols ∈ [9,10 ,11, 12] r=0.1

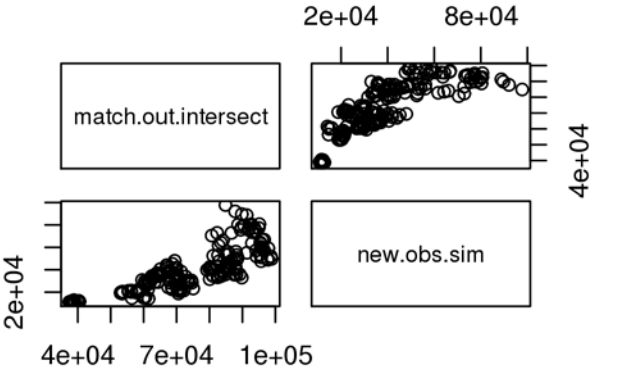


Abbildung 6.6 b ncols ∈ [9,10 ,11, 12] r=0.01

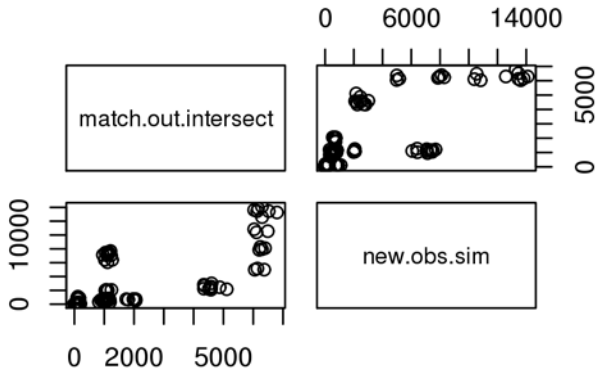


Abbildung 6.6 c ncols ∈ [3,4 ,5, 6] r=0.1

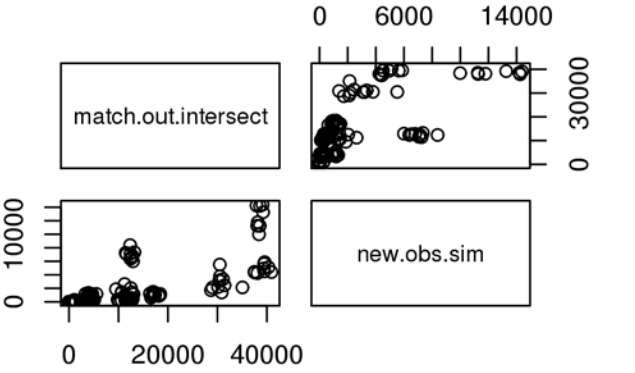


Abbildung 6.6 d ncols ∈ [9,10 ,11, 12] r=0.01

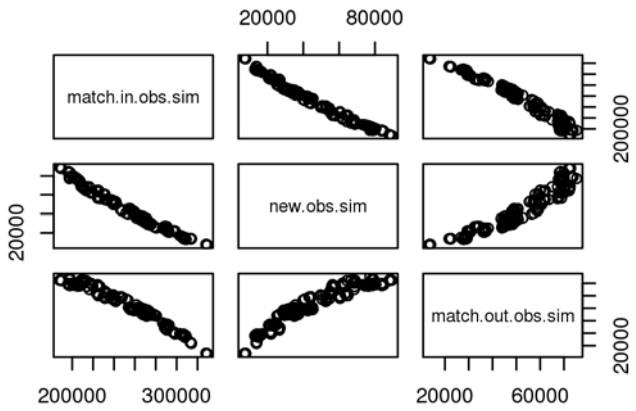


Abbildung 6.7 a ncols ∈ [9,10 ,11, 12] r=0.1

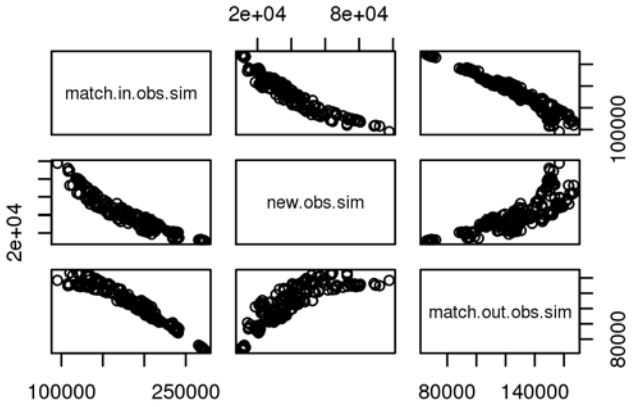


Abbildung 6.7 b ncols ∈ [9,10 ,11, 12] r=0.01

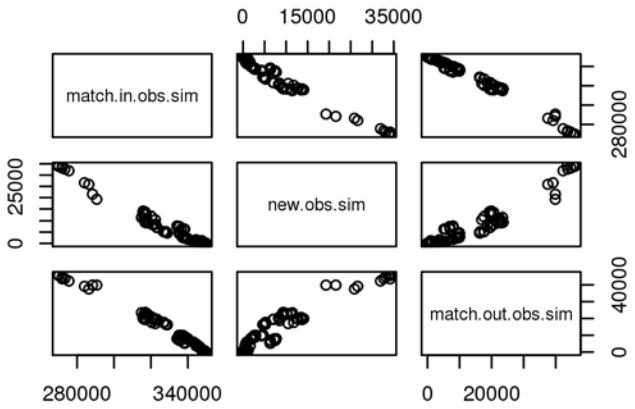


Abbildung 6.7 c ncols ∈ [3,4 ,5, 6] r=0.1

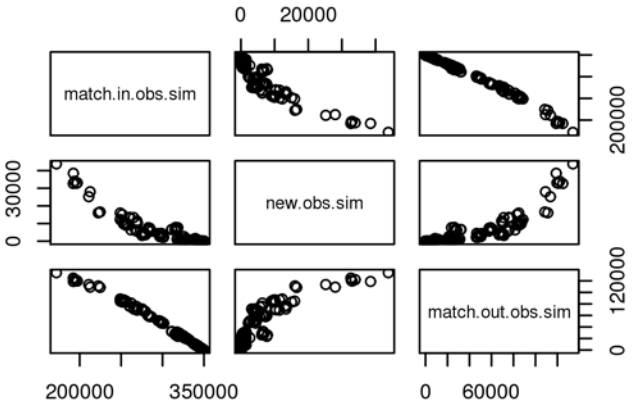


Abbildung 6.7 a ncols ∈ [3,4 ,5, 6] r=0.01

Weighted:

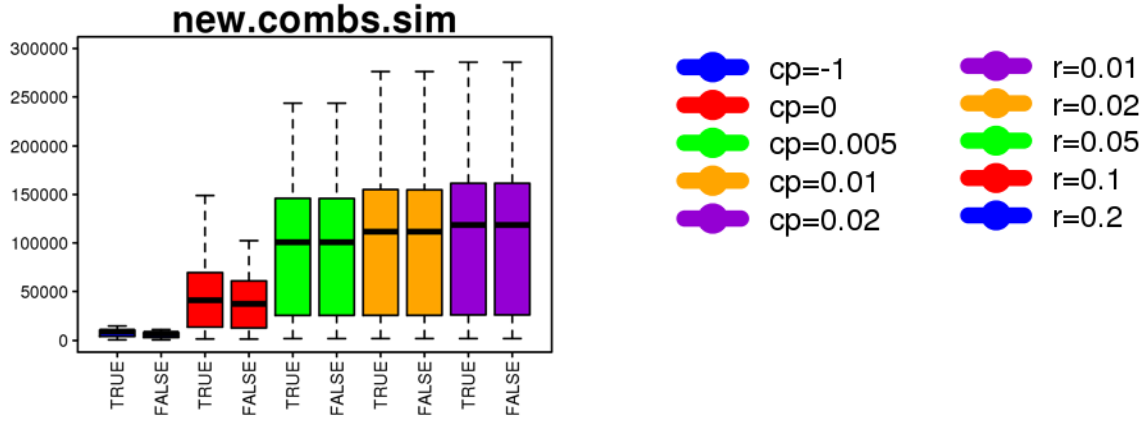


Abbildung 7.1 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

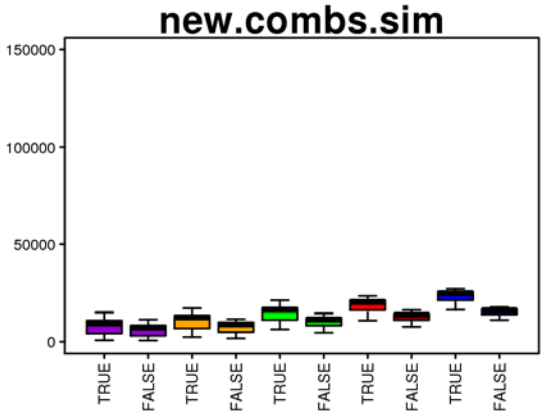


Abbildung 7.1 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

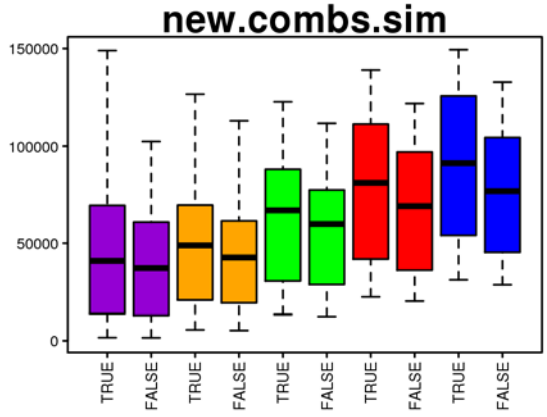


Abbildung 7.1 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

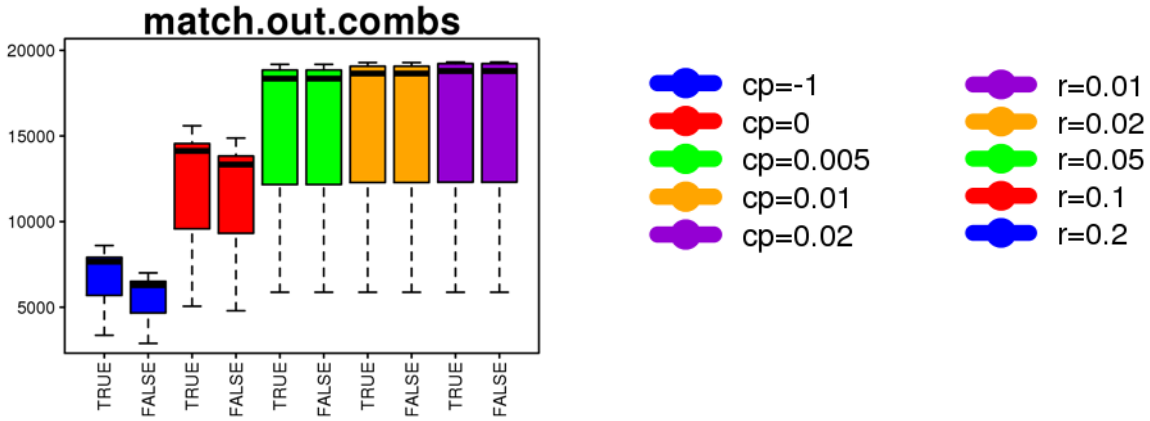


Abbildung 7.2 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

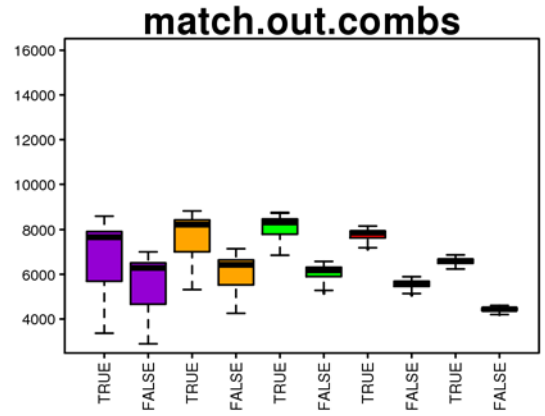


Abbildung 7.2 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

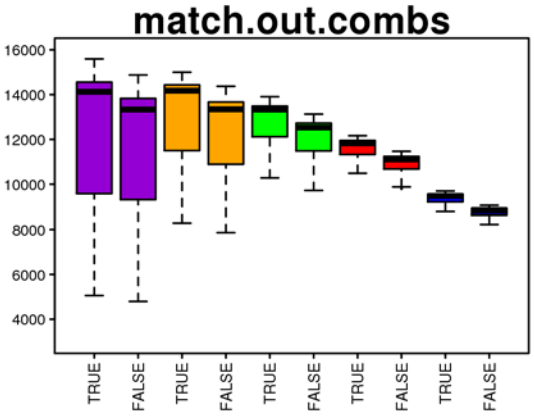


Abbildung 7.2 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

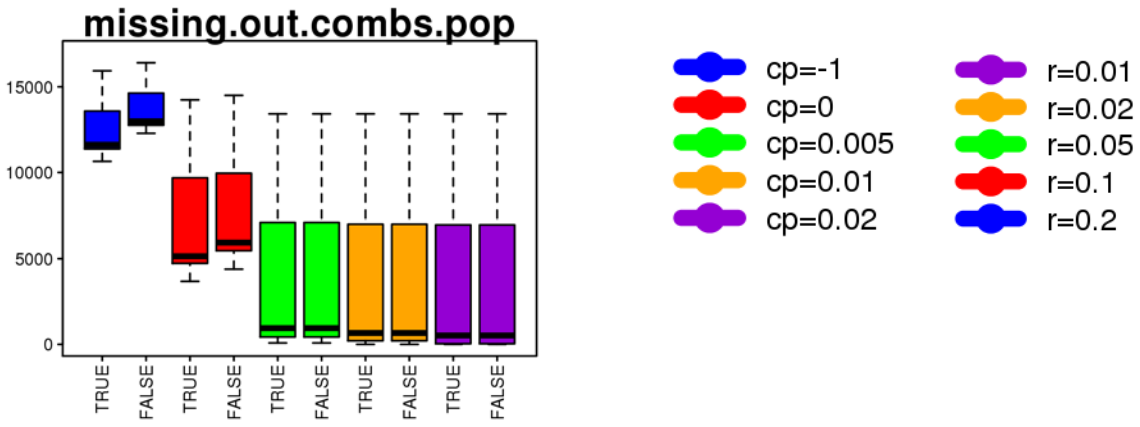


Abbildung 7.3 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

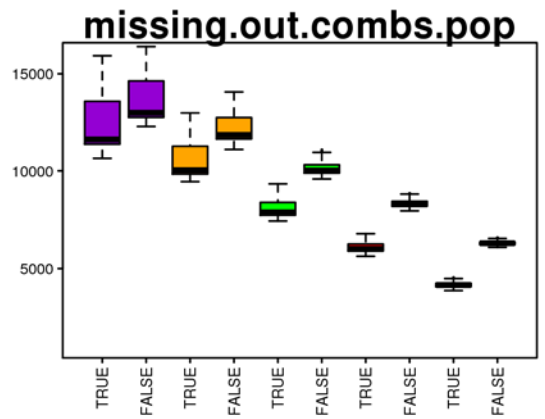


Abbildung 7.3 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

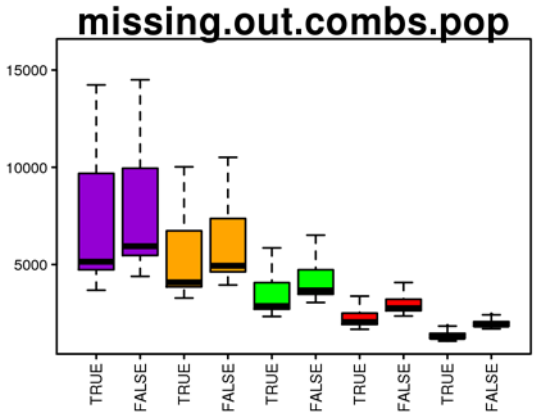


Abbildung 7.3 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

Weighted=TRUE: Es werden mehr neue, nicht in der realen Population auftretende Kombinationen erzeugt (Abbildung 7.1), dafür werden aber auch mehr Kombinationen erzeugt, die nicht in der Stichprobe waren, aber in der realen Population existieren (Abbildung 7.2). Dies wirkt sich auch auf Anzahl nicht erzeugter Populationskombinationen aus, die geringer ausfällt (Abbildung 7.3).

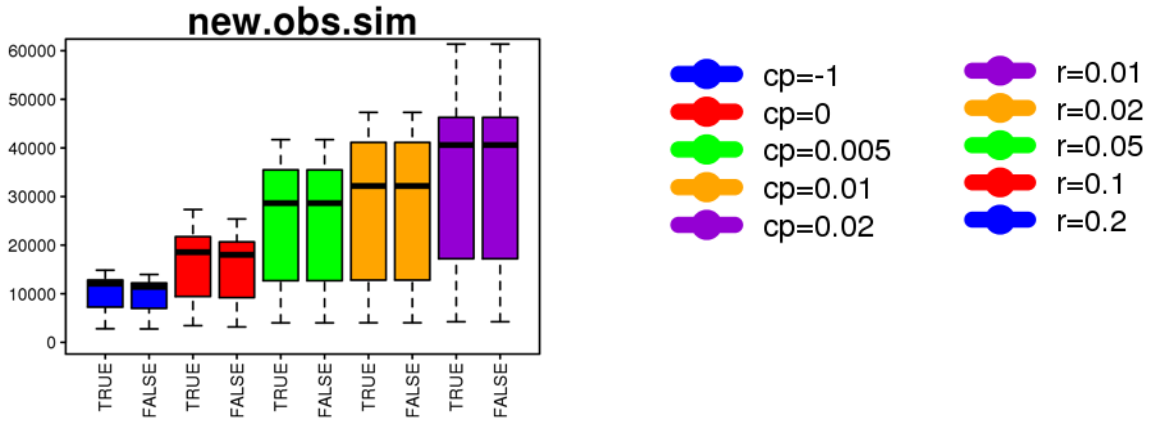


Abbildung 7.4 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

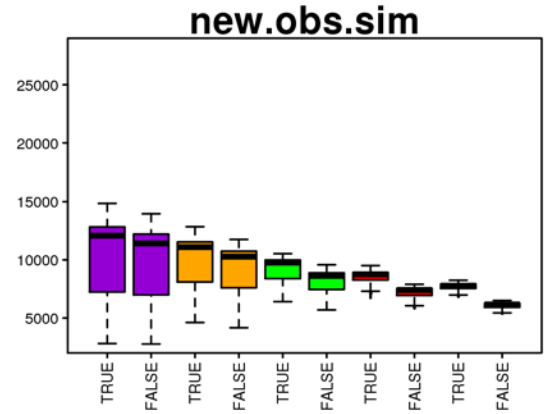


Abbildung 7.4 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

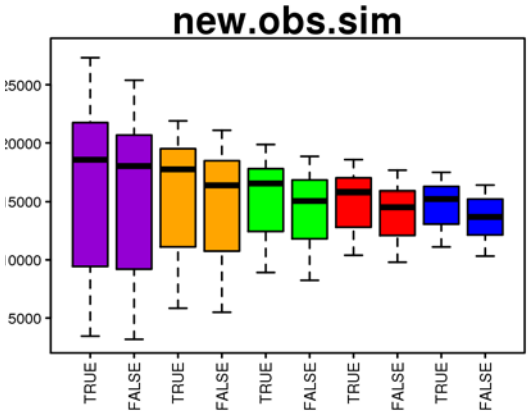
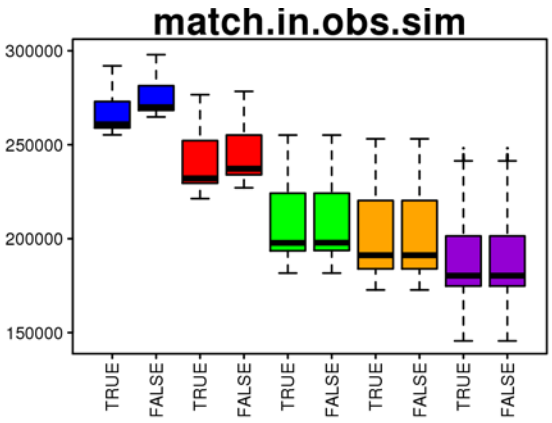


Abbildung 7.4 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]



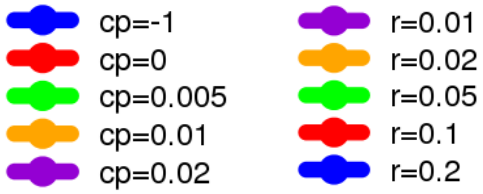


Abbildung 7.5 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

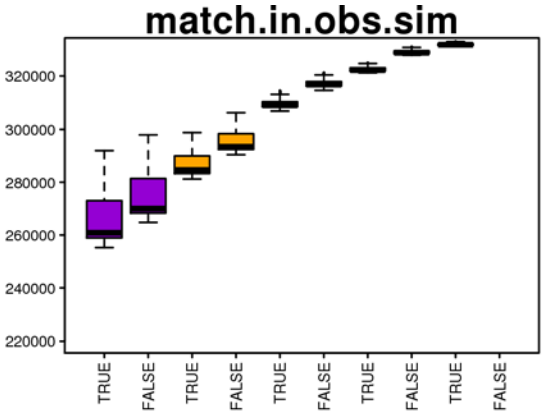


Abbildung 7.5 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

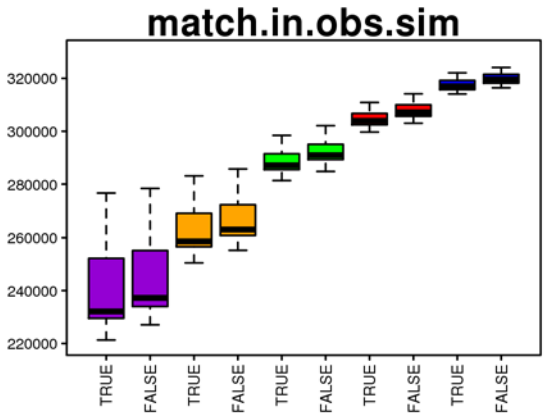
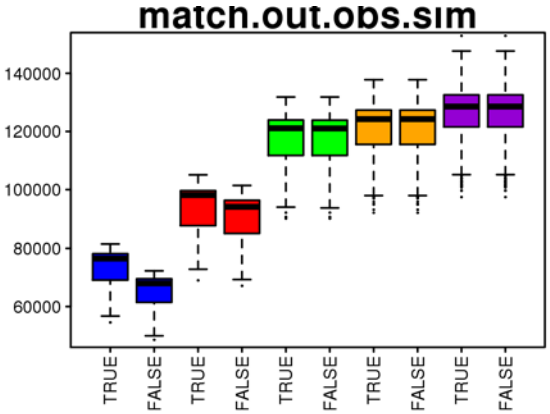


Abbildung 7.5 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]



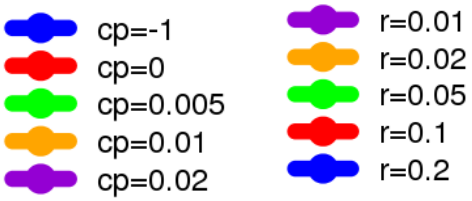


Abbildung 7.6 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

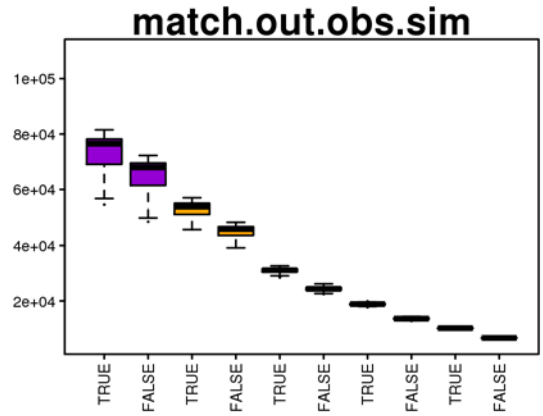


Abbildung 7.6 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

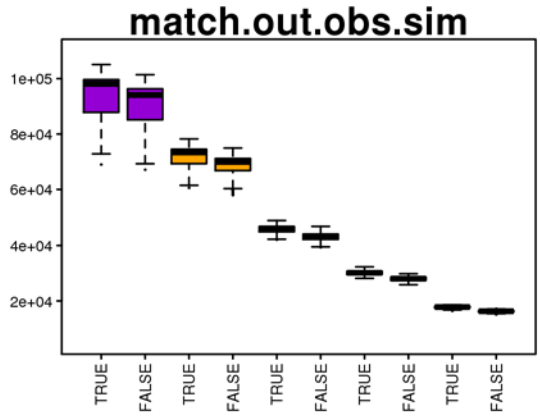
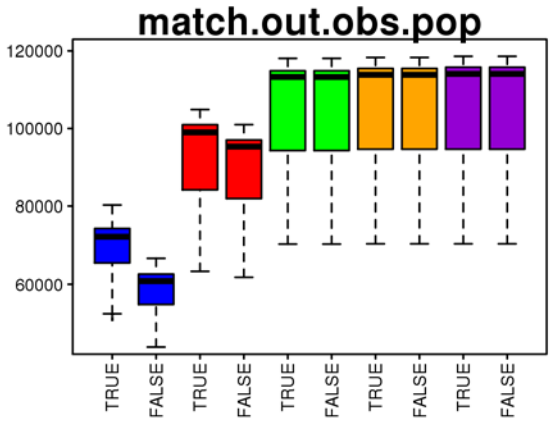


Abbildung 7.6 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

Weighted=TRUE: Wie bei den Kombinationen werden mehr Agenten synthetisiert, die nicht in der realen Population existieren (Abbildung 7.4). In der synthetischen Population gibt es weniger Agenten, deren Attributenkombination aus der Stichprobe entnommen wurde (Abbildung 7.5), aber mehr von denen, deren Kombination nicht direkt aus der Stichprobe stammt, aber in der realen Population vorkommt (Abbildung 7.6).



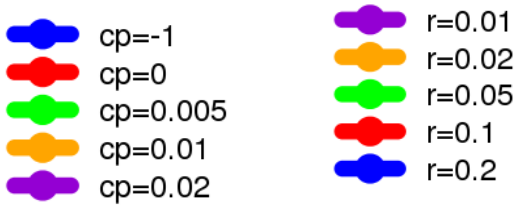


Abbildung 7.7 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

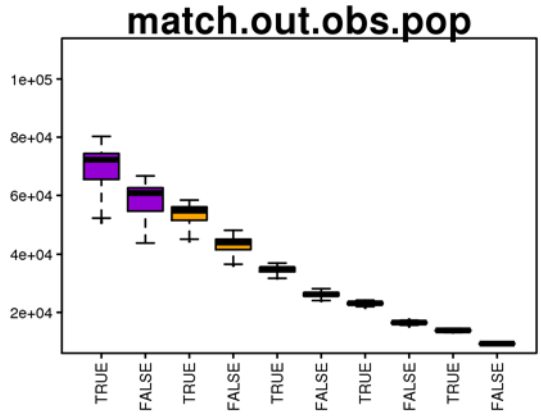


Abbildung 7.7 c cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

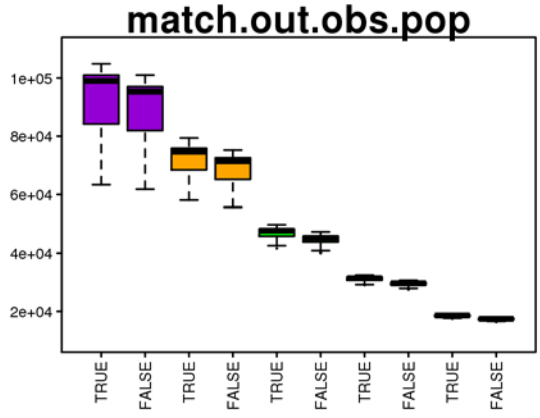


Abbildung 7.7 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

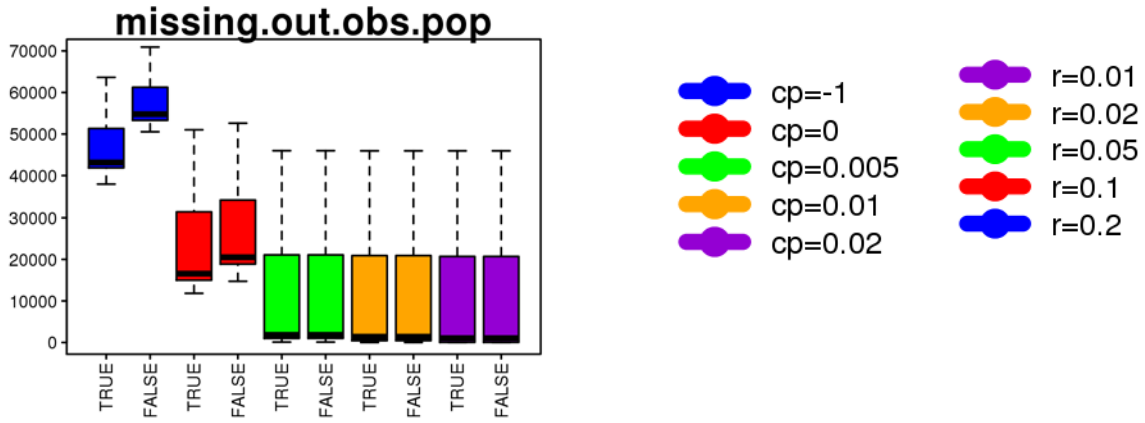


Abbildung 7.8 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

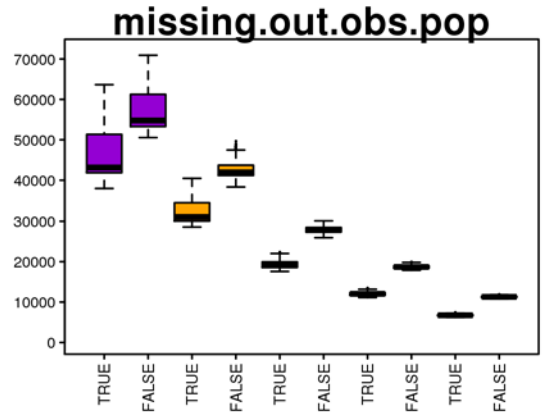


Abbildung 7.8 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

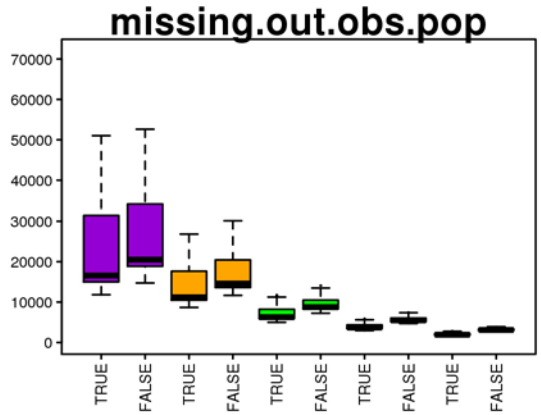


Abbildung 7.8 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

Weighted=TRUE: Die synthetische Population weist eine grössere Anzahl Agenten auf, deren Attributenkombination in der realen Population vorkommt, aber nicht in der Stichprobe zu finden ist (Abbildung 7.7) und es fehlen weniger Agenten, deren Kombination nicht in der Stichprobe ist (Abbildung 7.8).

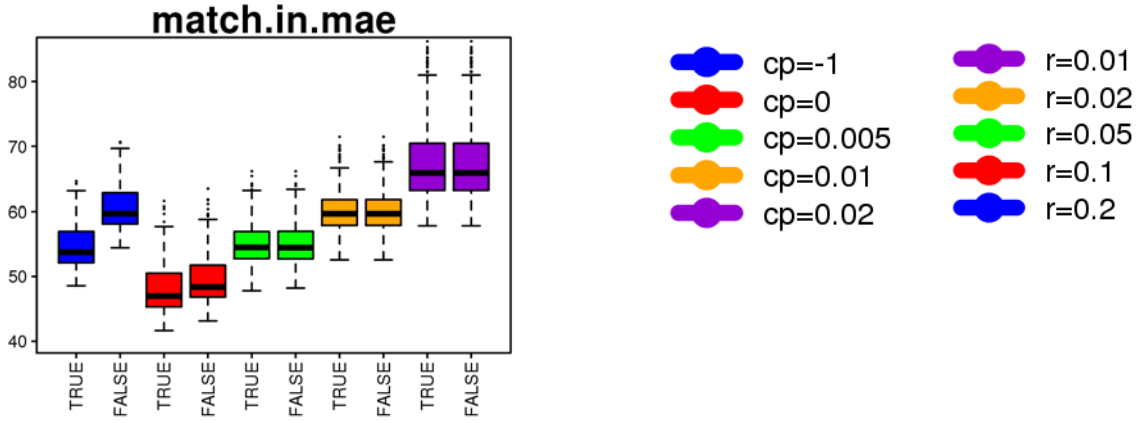


Abbildung 7.9 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

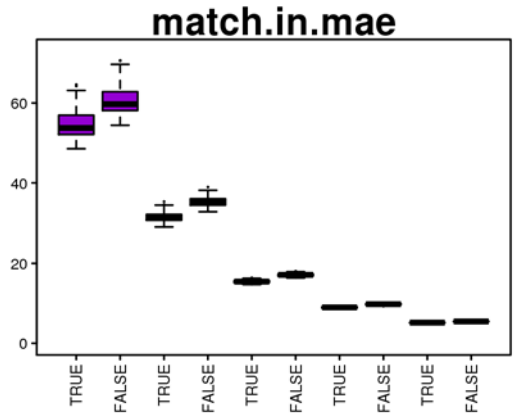


Abbildung 7.9 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

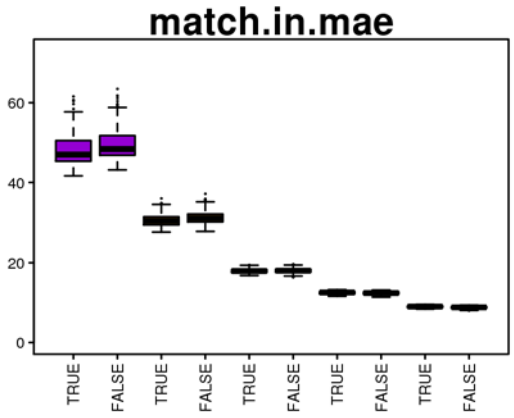
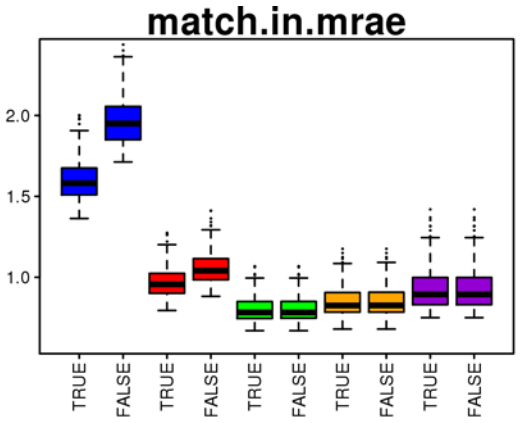


Abbildung 7.9 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]



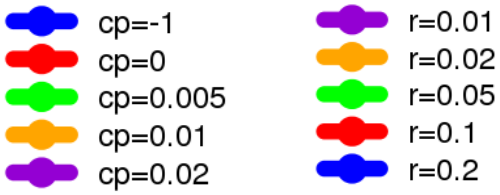


Abbildung 7.10 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

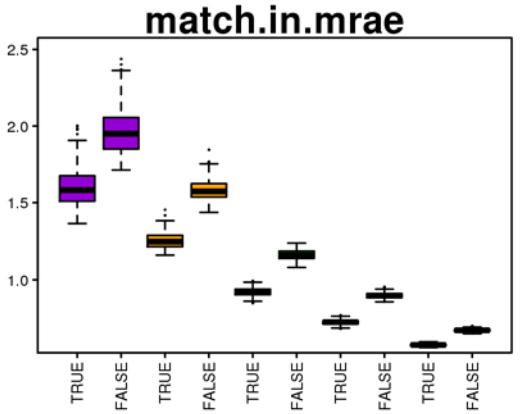


Abbildung 7.10 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

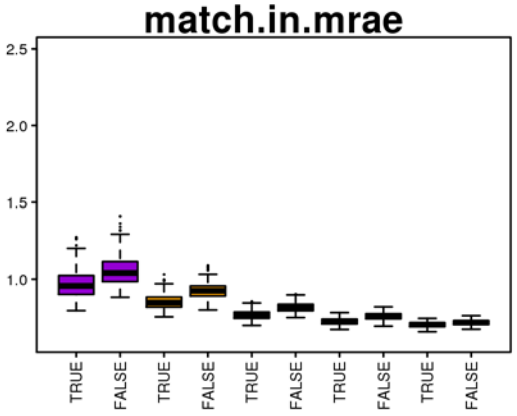
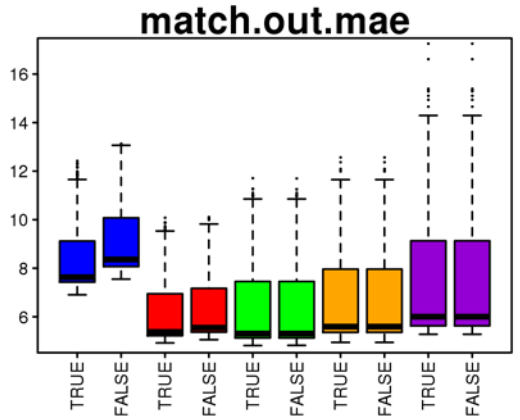


Abbildung 7.10 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]



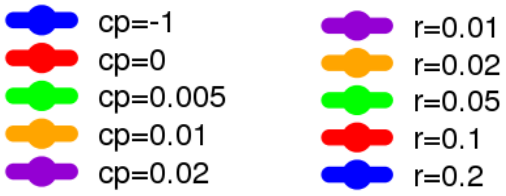


Abbildung 7.11 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

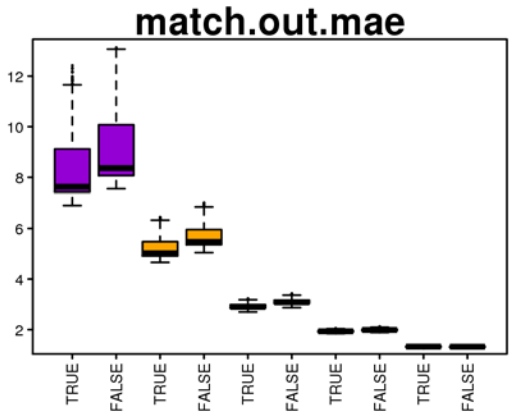


Abbildung 7.11 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

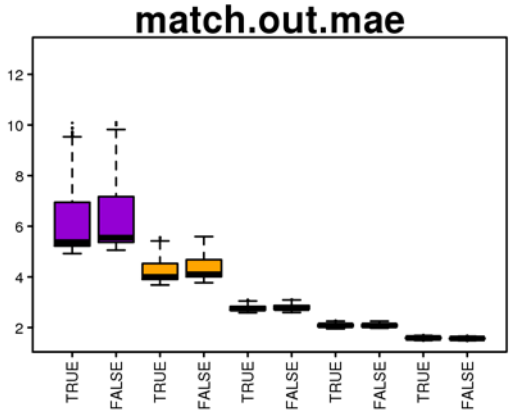
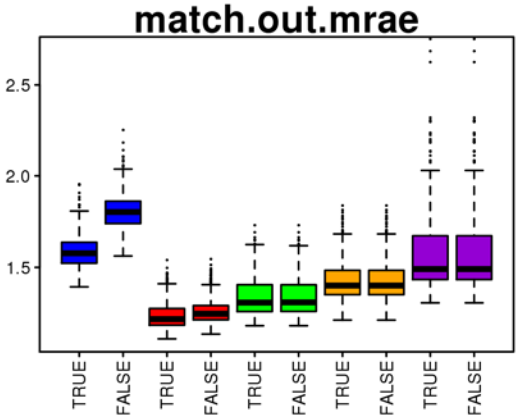


Abbildung 7.11 b cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]



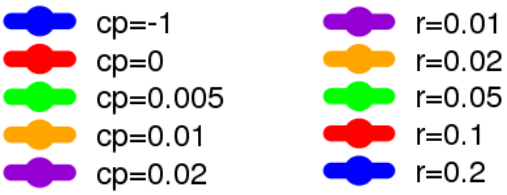


Abbildung 7.12 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

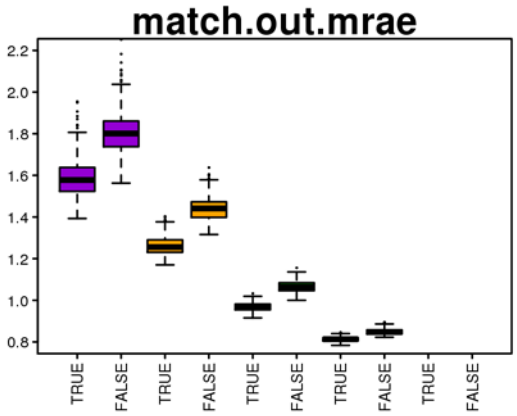


Abbildung 7.12 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

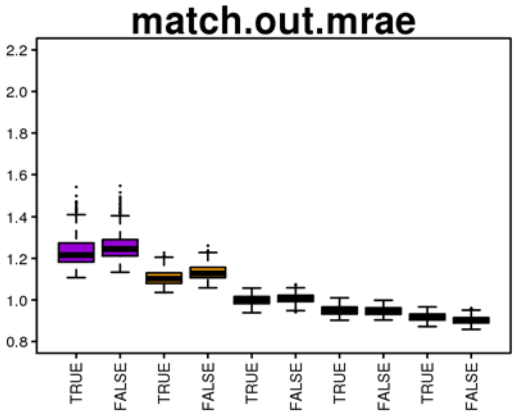


Abbildung 7.12 b cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

Weighted=TRUE: Die erzeugte Population repräsentiert die reale Population präziser. Dies wird durch einen geringeren absoluten und relativen Fehler festgestellt. Dieser ist für die beiden Agentenmengen des Typs match.in und match.out kleiner (Abbildungen 7.9, 7.10, 7.11, 7.12). Dies wird durch die Kulback-Leibler-Indikatoren partiell bestätigt. Der K.L-Indikator der Verteilungsfunktion der Agenten, deren Kombination in der Stichprobe auftaucht, weist einen niedrigeren Wert auf (Abbildung 7.13). Das Gegenteil ist bei dem K.L-Indikator der Verteilungsfunktion der Agenten der Fall, deren Attributenkombination in der realen Population enthalten ist, aber nicht in der Stichprobe (Abbildung 7.14).

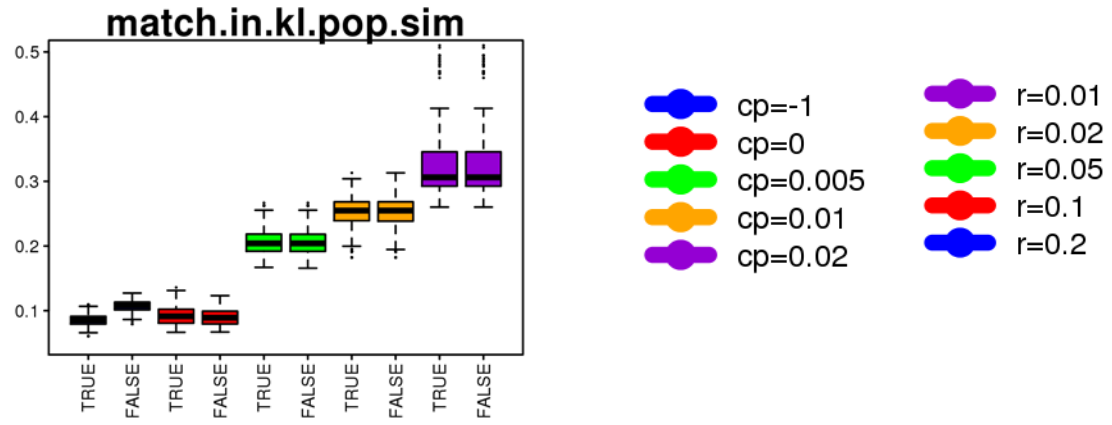


Abbildung 7.13a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

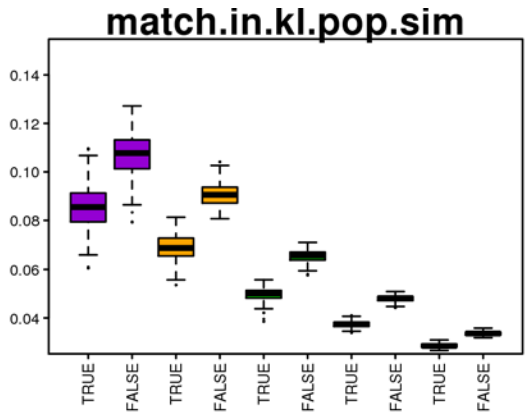


Abbildung 7.13 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

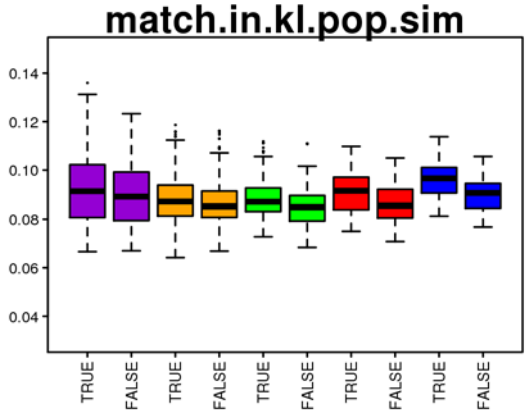


Abbildung 7.13 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

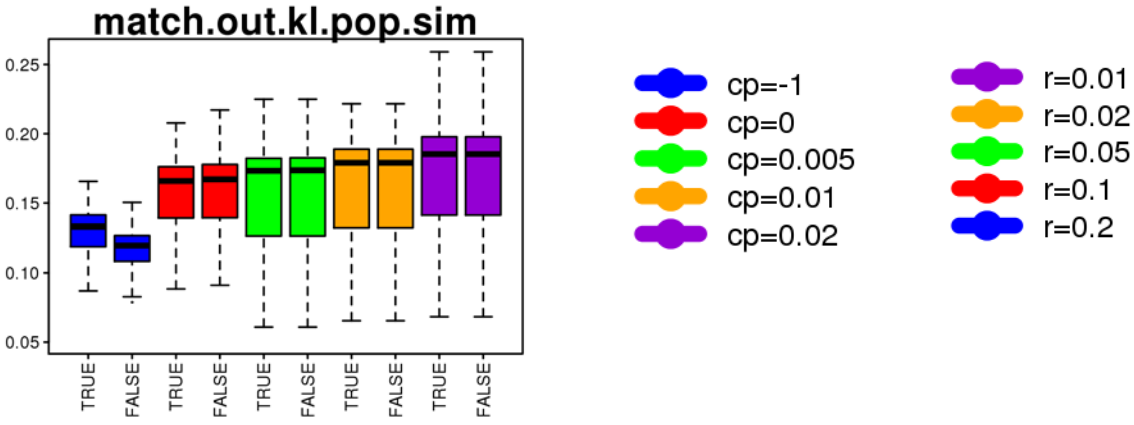


Abbildung 7.14 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

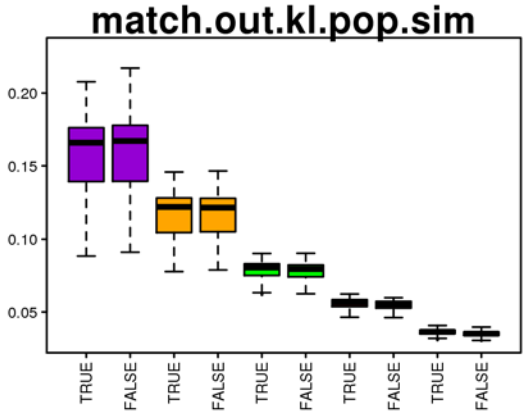


Abbildung 7.14 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

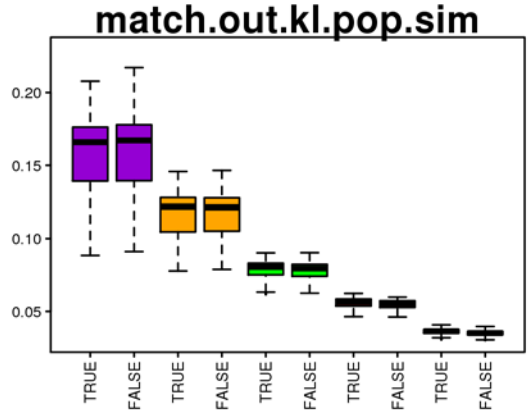


Abbildung 7.14 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

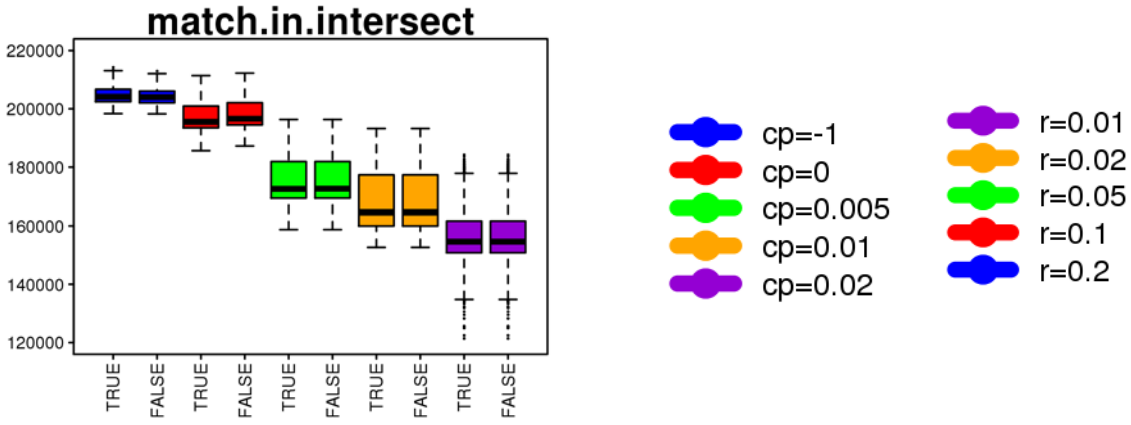


Abbildung 7.15a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

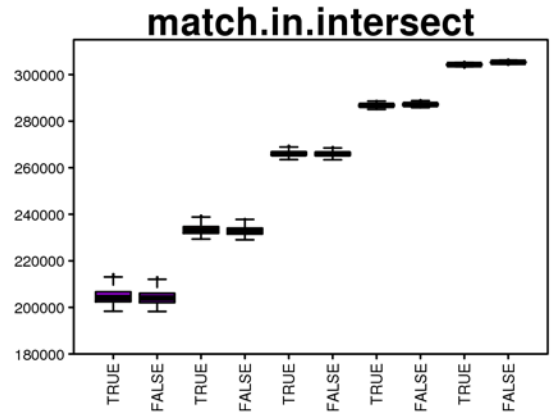


Abbildung 7.15 b cp=-1 r ∈[0.01; 0.02;0.05;0.1;0.2]

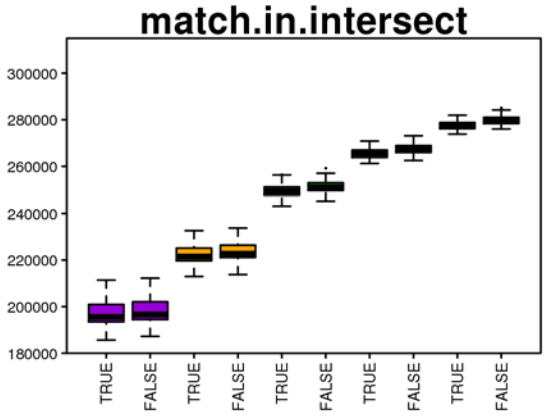


Abbildung 7.15 c cp=-1 r ∈[0.01; 0.02;0.05;0.1;0.2]

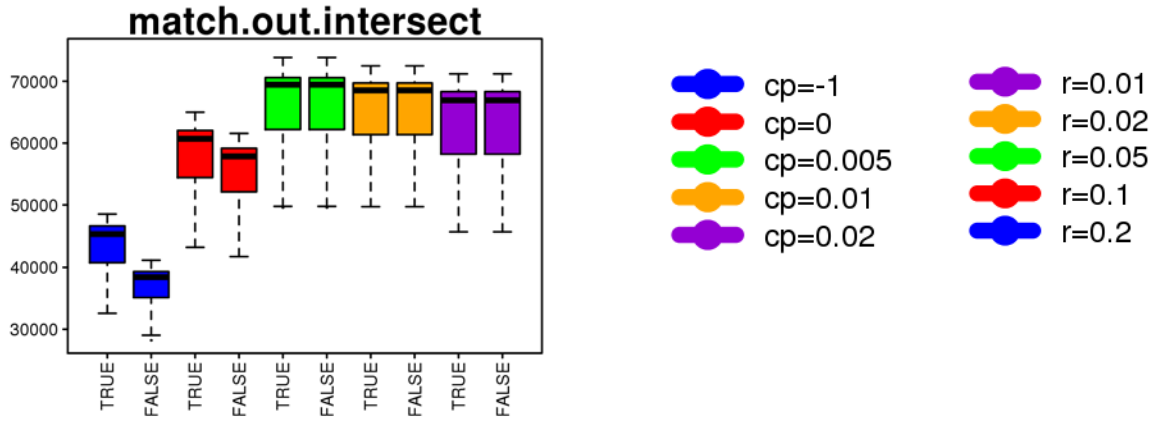


Abbildung 7.16 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

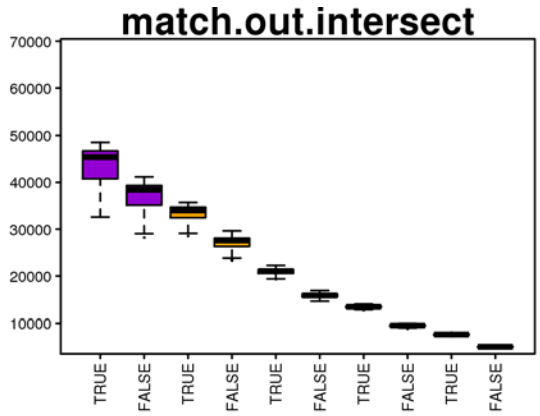


Abbildung 7.16 b cp=-1 r ∈[0.01; 0.02; 0.05; 0.1; 0.2]

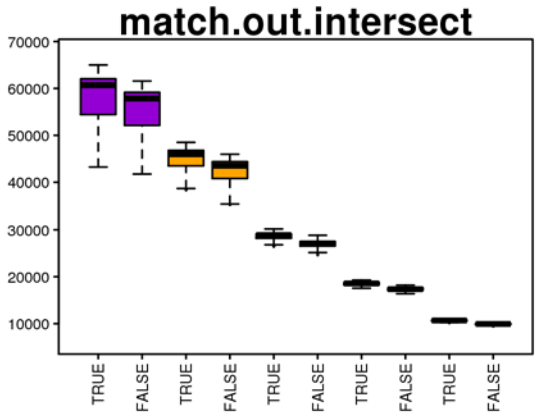


Abbildung 7.16 c cp=0 r ∈[0.01; 0.02; 0.05; 0.1; 0.2]

Die synthetische Population, die mittels weighted=TRUE erzeugt wird, besitzt eine grössere Anzahl Agenten, die denen der realen Population entsprechen und deren Kombination nicht in der Stichprobe zu sehen ist (Abbildung 7.16). Bei den erzeugten Agenten, deren Attributenkombination aus der Stichprobe stammt, und denen je ein realer Agent entspricht, ist kein grosser Unterschied zu erkennen (Abbildung 7.15). Diese beiden Tatsachen führen dazu, dass die Durchschnittsmenge der simulierten und der realen Population mit weighted=TRUE grösser ausfällt (Abbildung 7.17).

Fazit: Der Gewichtungsparameter weighted sollte gleich TRUE gesetzt werden, da die so erzeugte synthetische Population mehr der realen ähnelt. Dies wird von einem höheren Wert von *match.intersect* und niedrigeren Werten der Fehlerindikatoren abgeleitet.

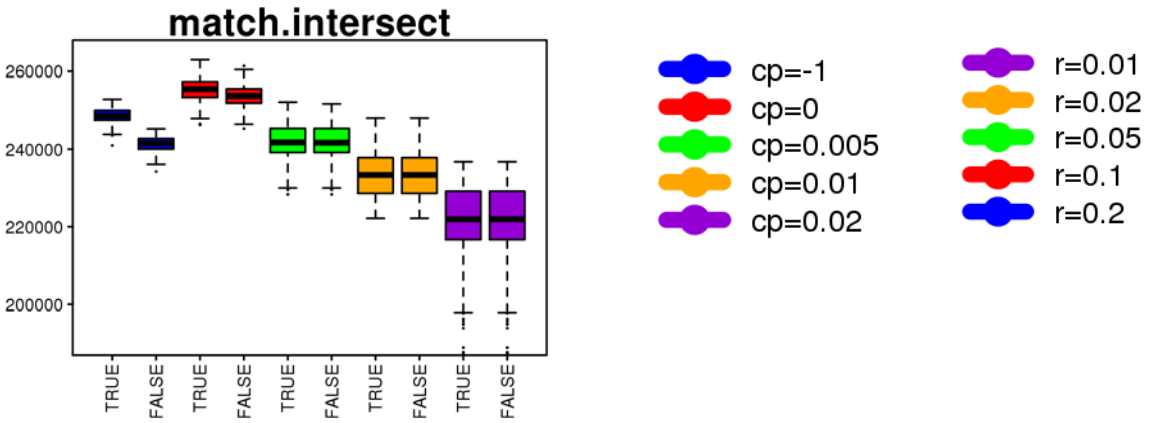


Abbildung 7.17 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

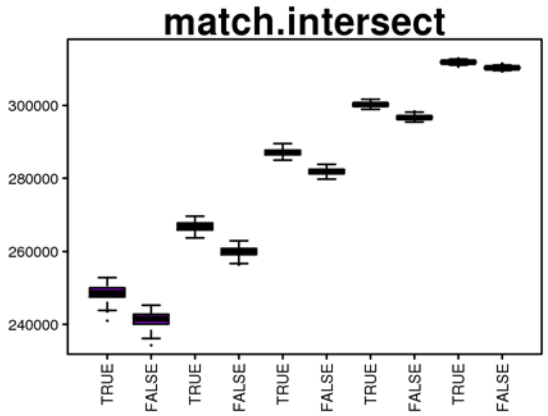


Abbildung 7.17 b cp=-1 r ∈[0.01;0.02;0.05;0.1;0.2]

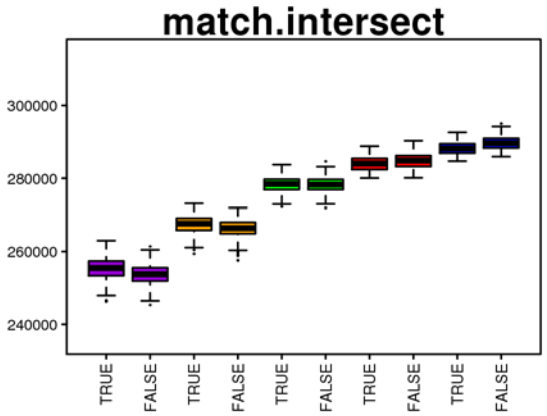


Abbildung 7.17 c cp=0 r ∈[0.01;0.02;0.05;0.1;0.2]

MM:

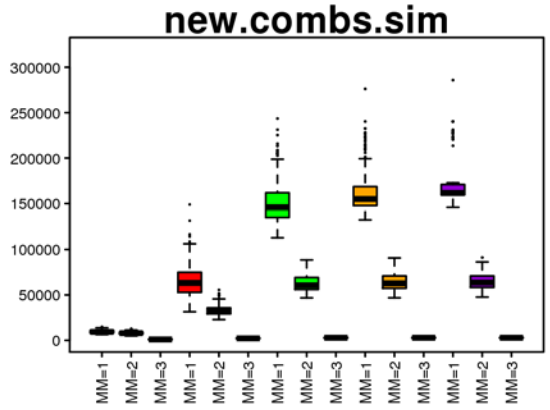


Abbildung 8.1 a r=0.01 cp ∈[-1; 0; 0.005; 0.01; 0.02]

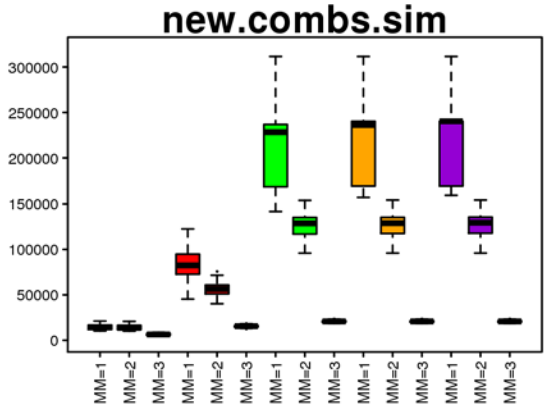


Abbildung 8.1 b r=0.05 cp ∈[-1; 0; 0.005; 0.01; 0.02]

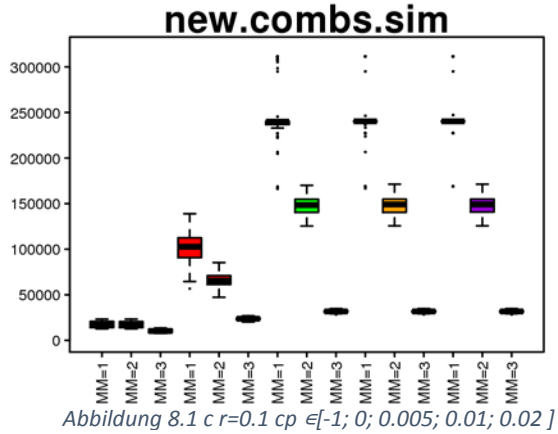
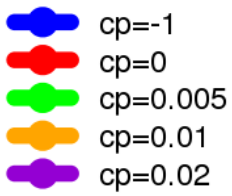


Abbildung 8.1 c r=0.1 cp ∈[-1; 0; 0.005; 0.01; 0.02]



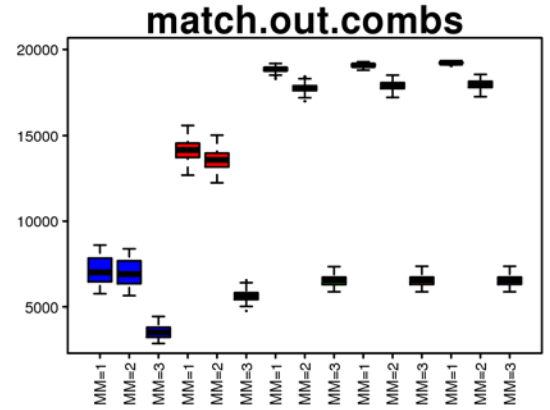


Abbildung 8.2 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

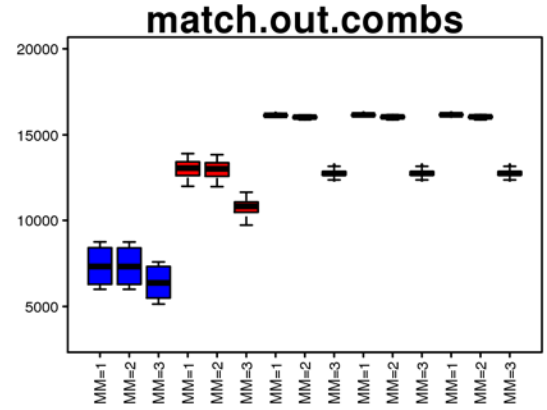
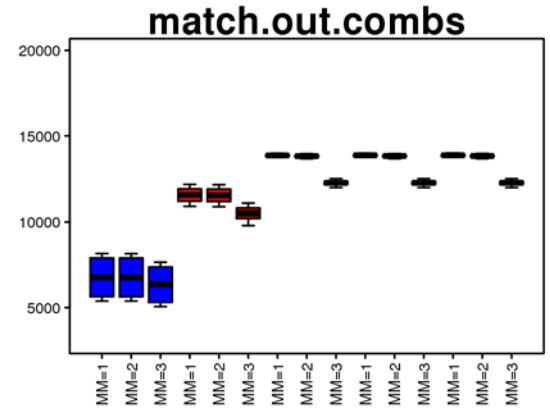


Abbildung 8.2 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



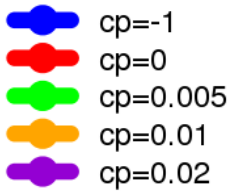


Abbildung 8.2 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

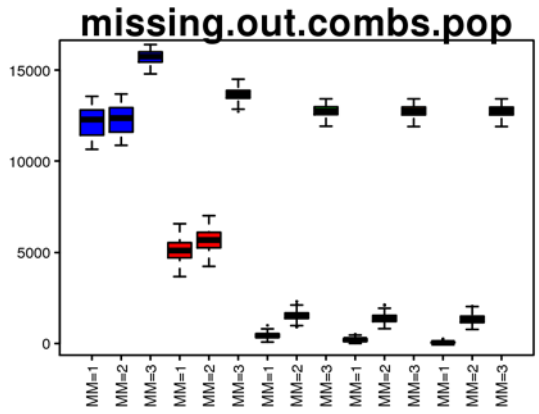


Abbildung 8.3 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

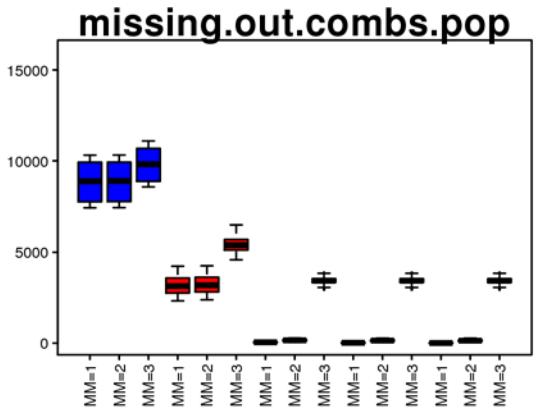


Abbildung 8.3 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

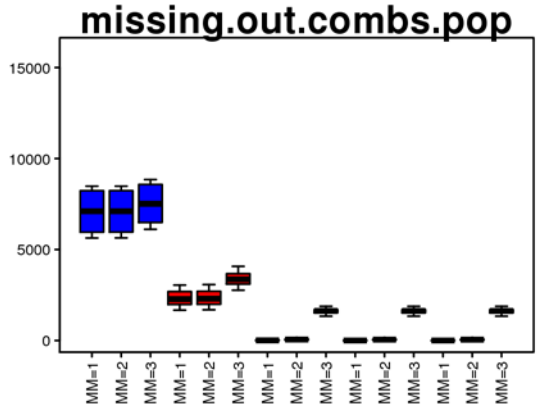
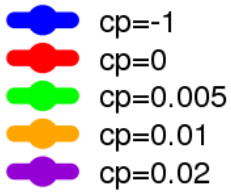


Abbildung 8.3 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



Mit steigendem MM-Wert wird, wie zu erwarten, eine geringere Anzahl neuer Attributenkombinationen erzeugt, die nicht in der realen Population existieren. Dabei fällt der Unterschied zwischen MM=3 und MM=2 deutlich höher aus als zwischen MM=2 und MM=1 (Abbildung 8.1). Das Gegenteil ist bei den nicht erzeugten Kombinationen zu beobachten, die nicht in der Stichprobe sind, aber in der realen Population existieren (Abbildung 8.2). Bei MM=3 fehlen im Vergleich zu MM=1 und MM=2 deutlich mehr von diesen Kombinationen. Dies hat zur Folge, dass bei MM=3 die meisten Attributenkombinationen, die nicht in der Stichprobe auftauchen, im Vergleich zur realen Population fehlen (Abbildung 8.3). Es ist ein grosser Unterschied der Grössenordnung dieses Phänomens bei einer kleinen Stichprobe bemerkbar.

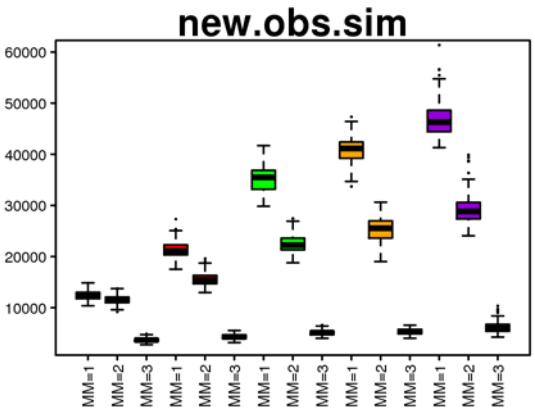


Abbildung 8.4 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

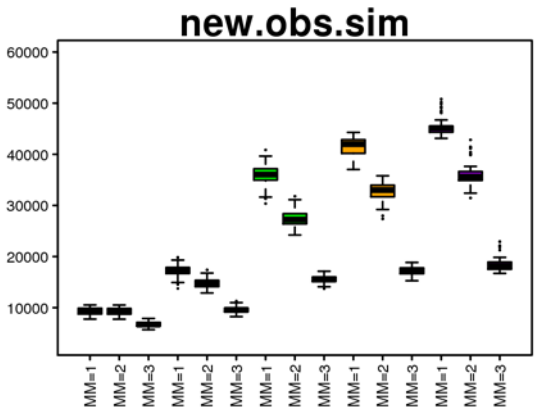


Abbildung 8.4 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

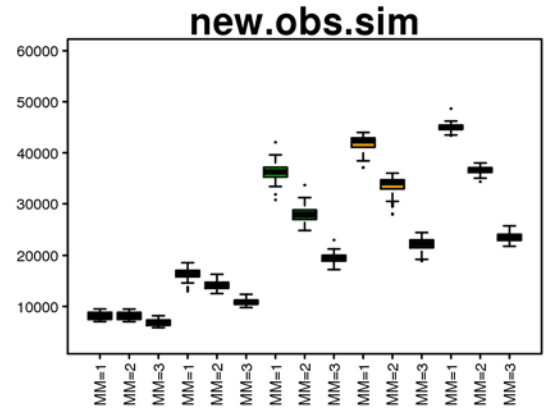
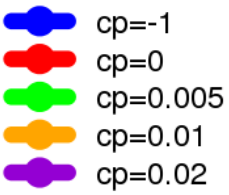


Abbildung 8.4 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



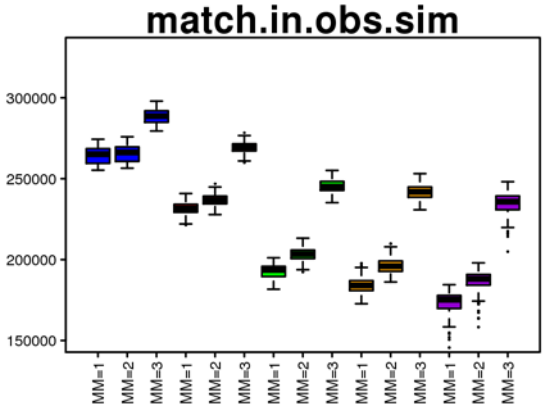


Abbildung 8.5 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

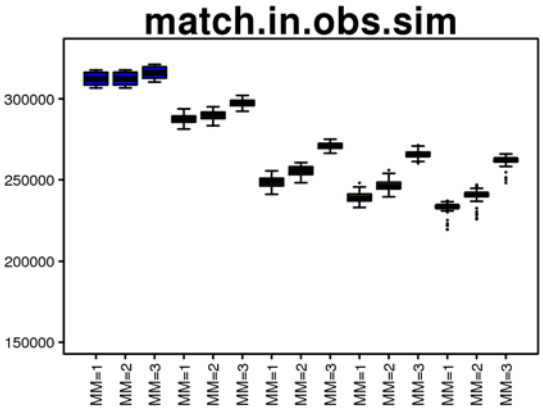


Abbildung 8.5 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

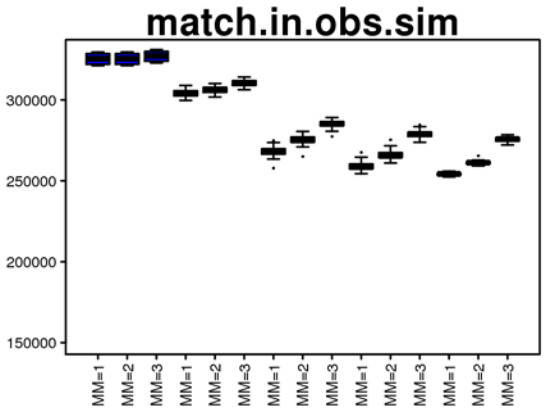
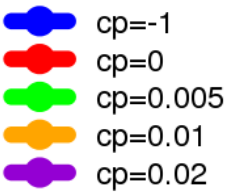


Abbildung 8.5 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



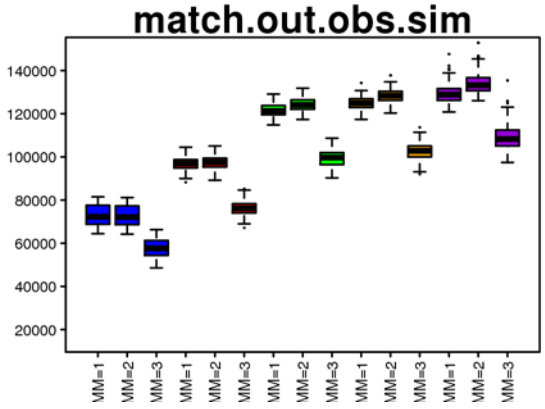


Abbildung 8.6 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

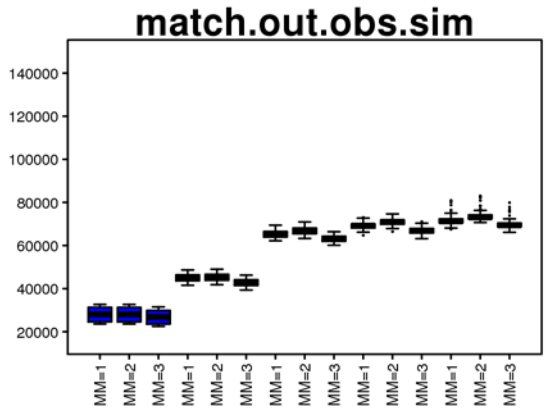
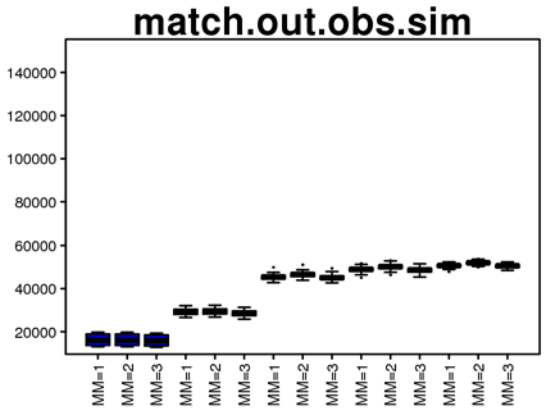


Abbildung 8.6 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



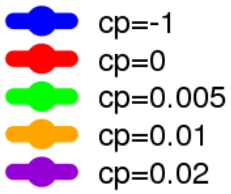


Abbildung 8.6 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Je grösser der MM-Wert gesetzt wird, desto weniger neue simulierte Agenten, deren Attributenkombination nicht in der realen Population vorhanden ist, werden erzeugt (Abbildung 8.4). Bei grösser werdendem cp-Wert tritt dieses Phänomen verstärkt auf, d.h. der Unterschied zwischen den neuerzeugten Agenten mit verschiedenen MM Werten wächst mit steigendem cp Wert. Die Anzahl erzeugter Agenten, deren Kombination in der Stichprobe enthalten ist, wird mit steigendem MM grösser (Abbildung 8.5). Das Gegenteil passiert mit den Agenten, deren Attributenkombination nicht der Stichprobe zu entnehmen ist (Abbildung 8.6). Diese Effekte sind ausgeprägter bei kleineren Stichproben und höheren cp-Werten.

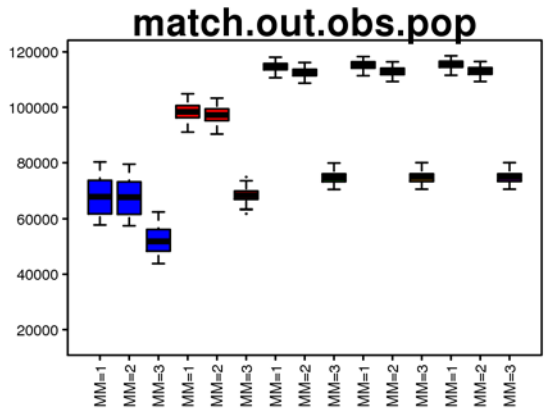


Abbildung 8.7 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

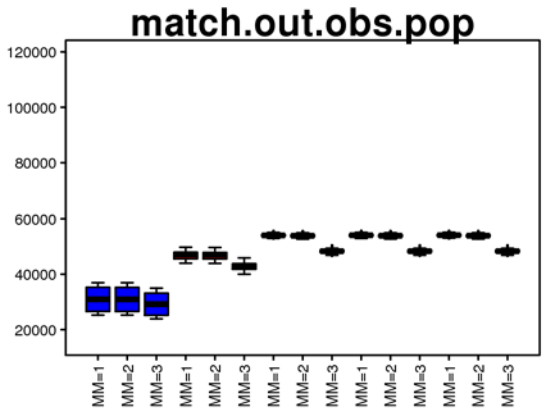


Abbildung 8.7 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

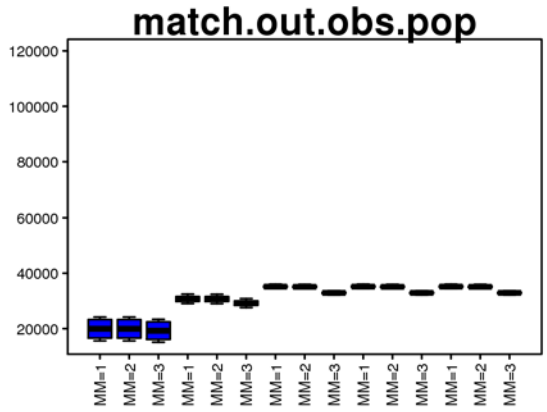
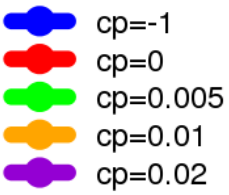


Abbildung 8.7 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



Die Werte der Kombinationsindikatoren bestätigen die Resultate der simulierten Observationsindikatoren. Je grösser MM gewählt wird, desto weniger Observationen, deren Attributenkombination nicht in der Stichprobe ist, werden in die synthetische Population miteinbezogen (Abbildung 8.7) und desto mehr werden gar nicht miteinbezogen (Abbildung 8.8). Auch hier verstärkt sich der Effekt mit sinkender Stichprobengrösse und steigendem cp-Wert.

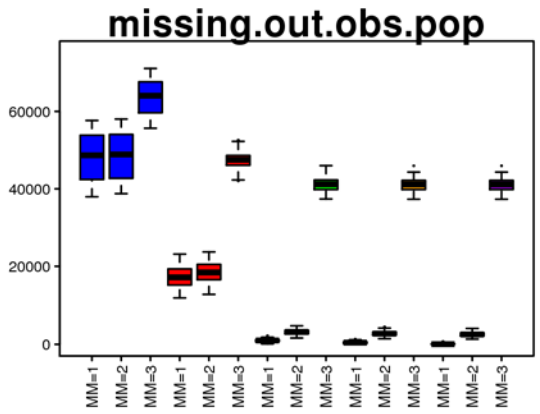


Abbildung 8.8 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

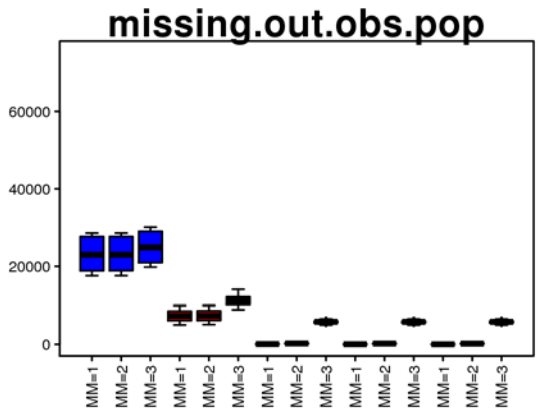


Abbildung 8.8 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

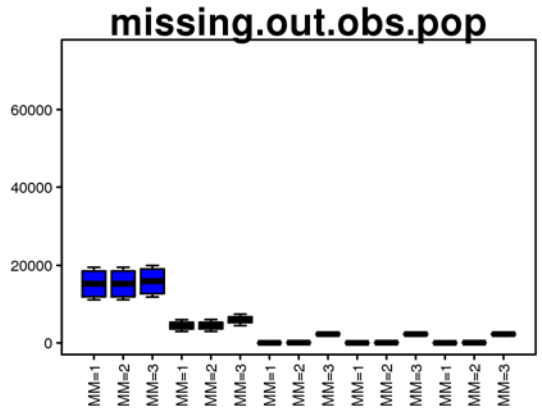
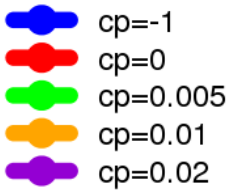


Abbildung 8.8 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



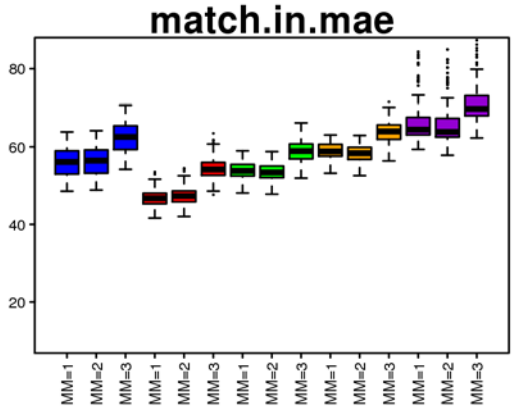


Abbildung 8.9 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

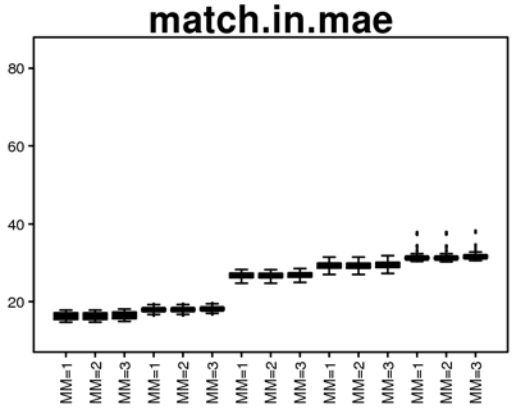


Abbildung 8.9 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

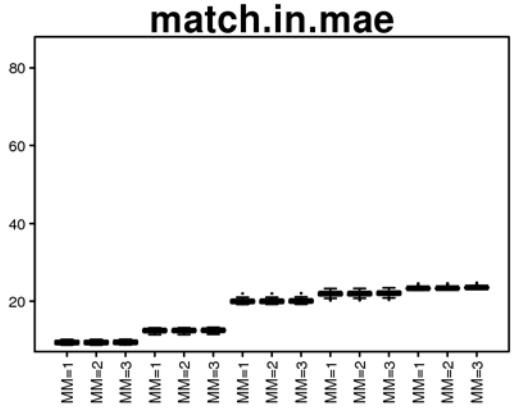
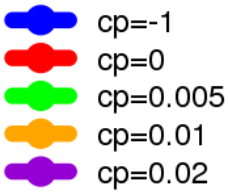


Abbildung 8.9 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



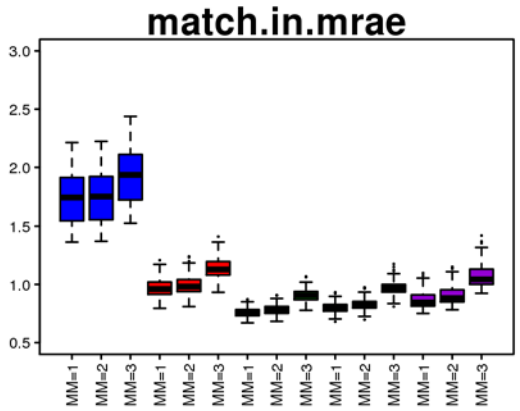


Abbildung 8.10 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

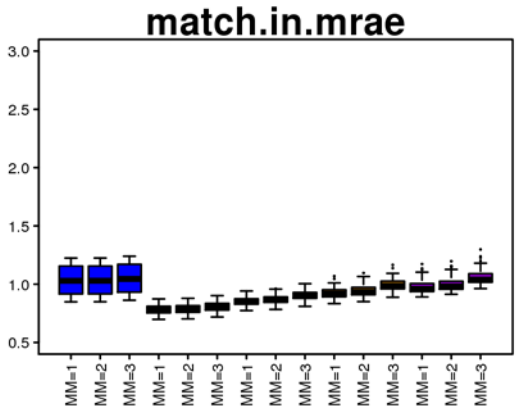
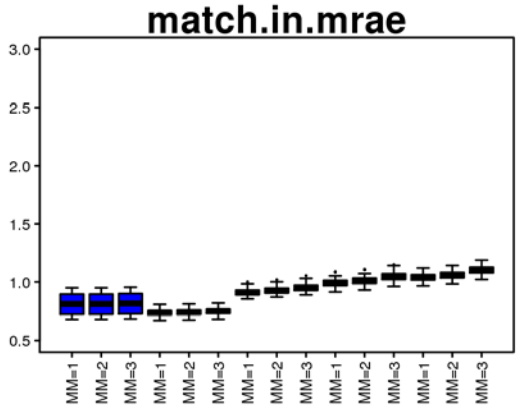


Abbildung 8.10 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



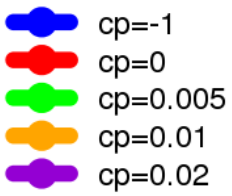


Abbildung 8.10 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

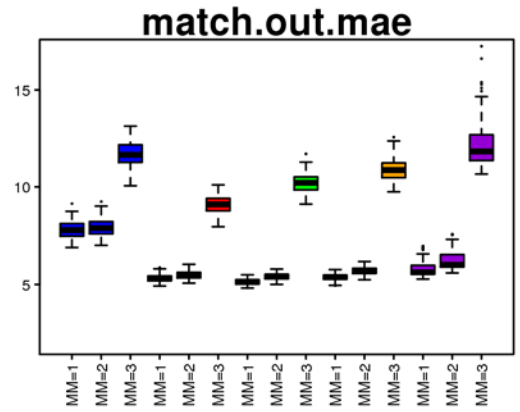


Abbildung 8.11 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

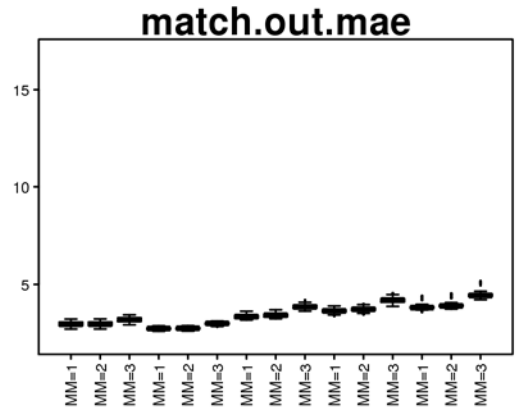
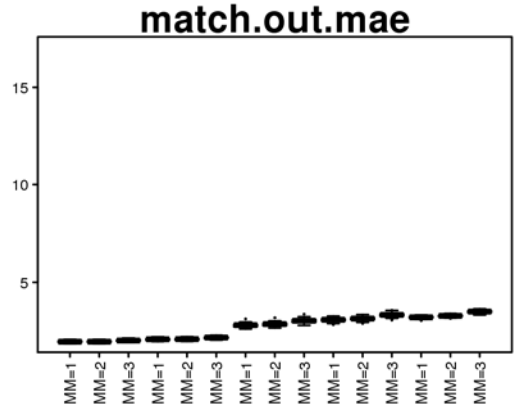


Abbildung 8.11 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



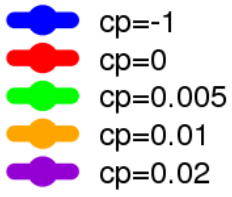


Abbildung 8.11 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

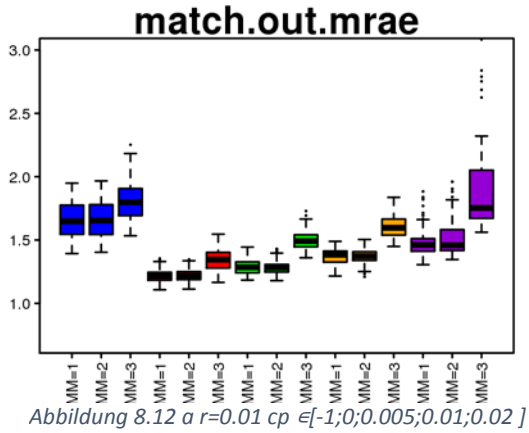


Abbildung 8.12 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

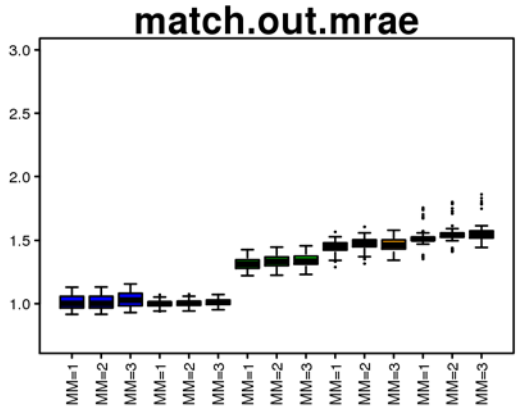
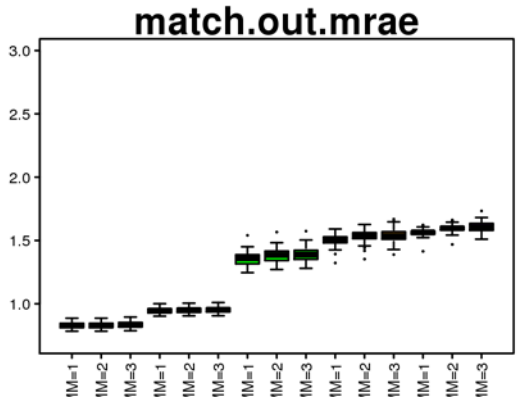


Abbildung 8.12 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



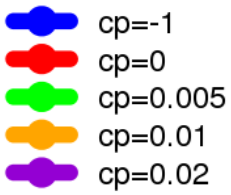


Abbildung 8.12 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Der absolute und der relative Fehler bei den Observationen, deren Kombination in der Stichprobe enthalten ist, wird mit steigendem MM grösser, wobei zwischen MM=2 und MM=3 der Sprung markanter ausfällt (Abbildungen 8.9, 8.10). Die Differenz verblasst mit steigender Stichprobengrösse. Dasselbe ist bei den Observationen, deren Kombination nicht in der Stichprobe zu finden ist, zu verzeichnen (Abbildungen 8.11, 8.12). Betrachtet man die K.L.-Divergenz der „in sample“ Agenten ist bei den verschiedenen MM-Werten kein relevanter Unterschied bemerkbar (Abbildung 8.13). Betrachtet man hingegen die synthetisierte Agentenmenge, deren Attributenkombination vom Algorithmus richtig erfasst wird, aber nicht in der Stichprobe auftaucht, werden bei steigendem MM niedrigere Werte festgestellt (Abbildung 8.14). Dieser Effekt nimmt bei grösser werdendem cp-Wert zu.

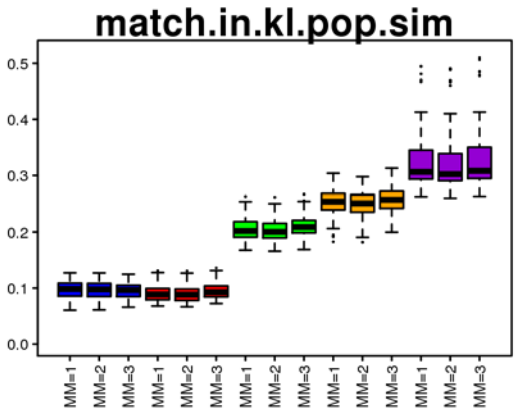


Abbildung 8.13 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

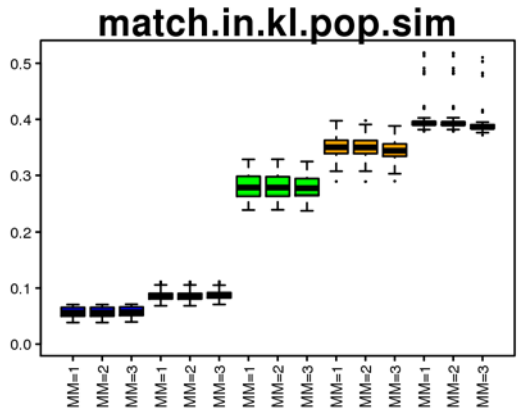
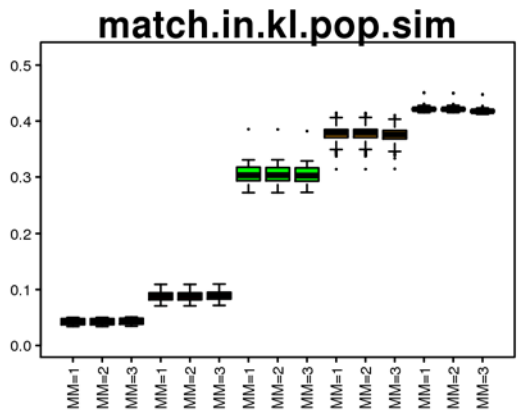


Abbildung 8.13 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



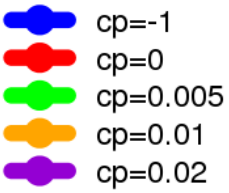


Abbildung 8.13 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

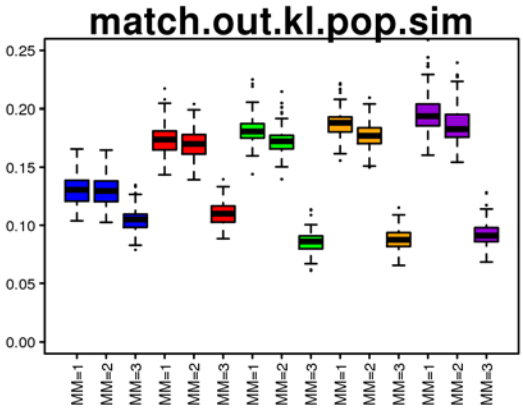


Abbildung 8.14 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

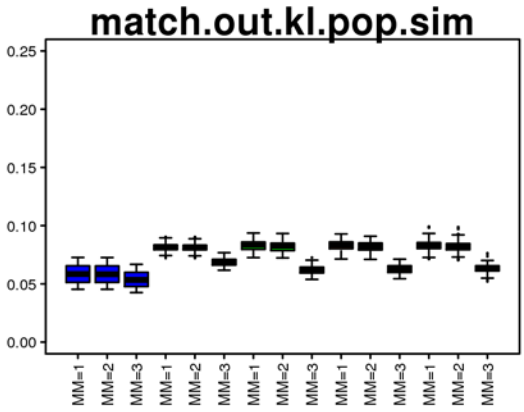
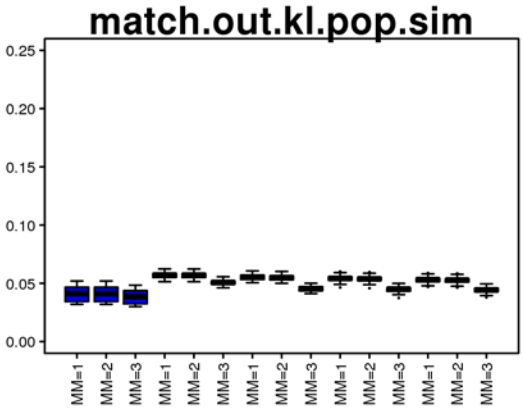


Abbildung 8.14 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



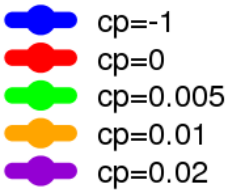


Abbildung 8.14 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

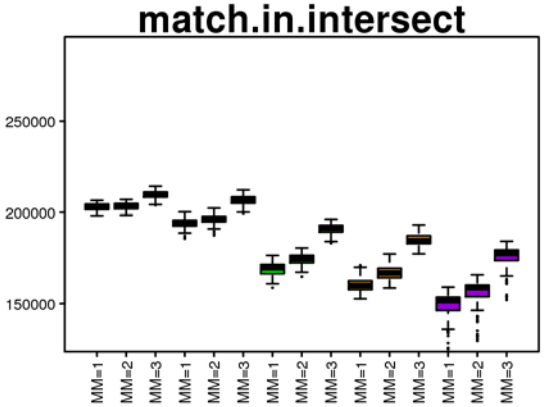


Abbildung 8.15 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

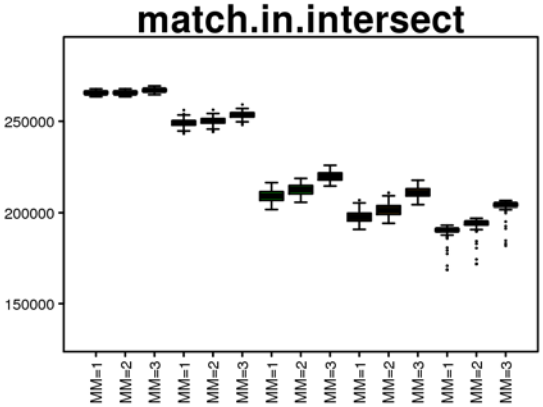
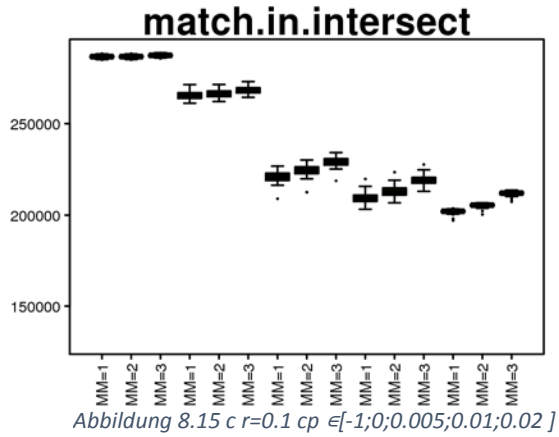


Abbildung 8.15 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



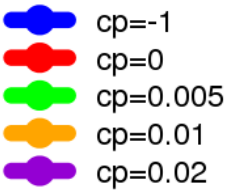


Abbildung 8.15 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

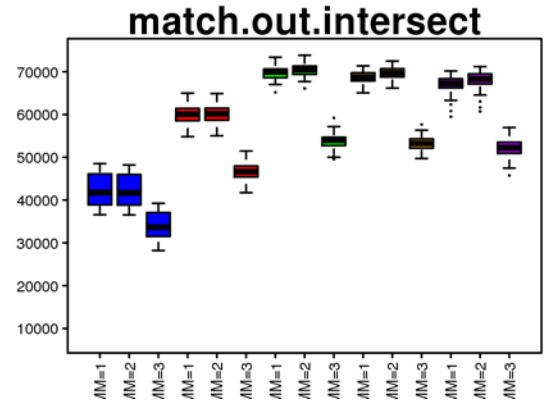


Abbildung 8.16 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

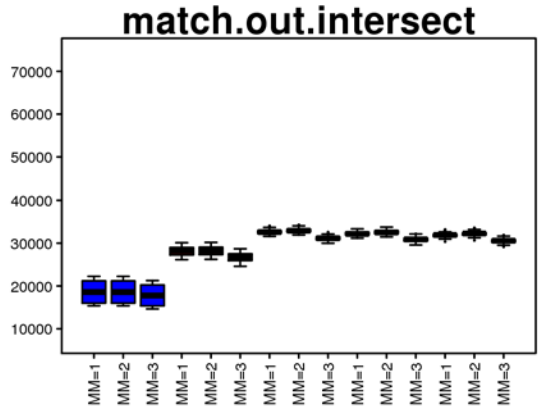
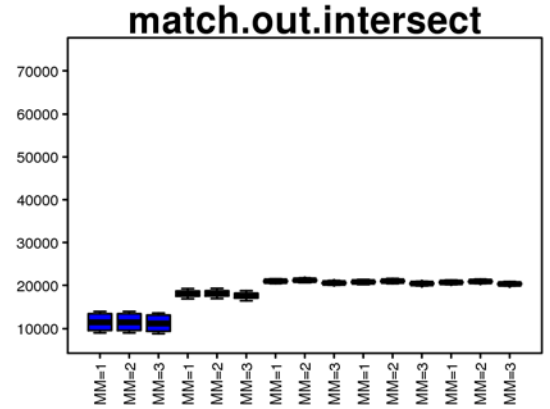


Abbildung 8.16 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



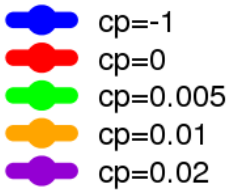


Abbildung 8.16 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

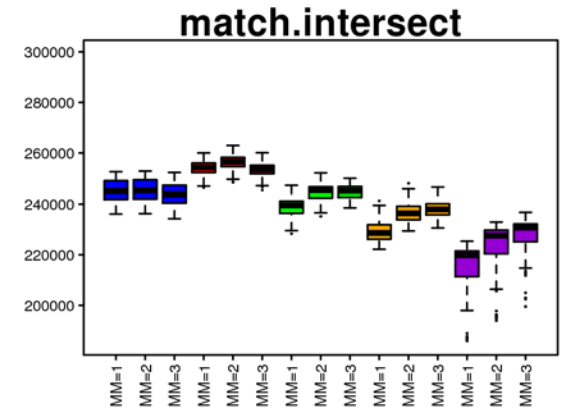


Abbildung 8.17 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

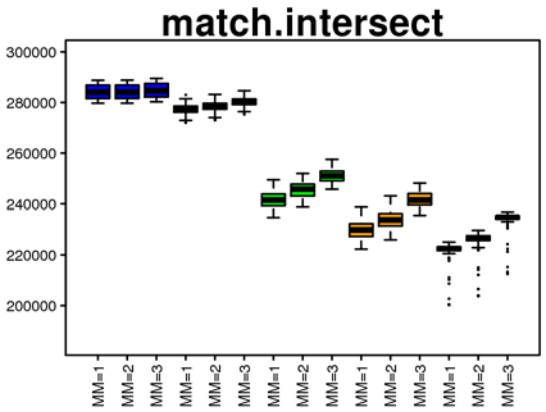


Abbildung 8.17 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

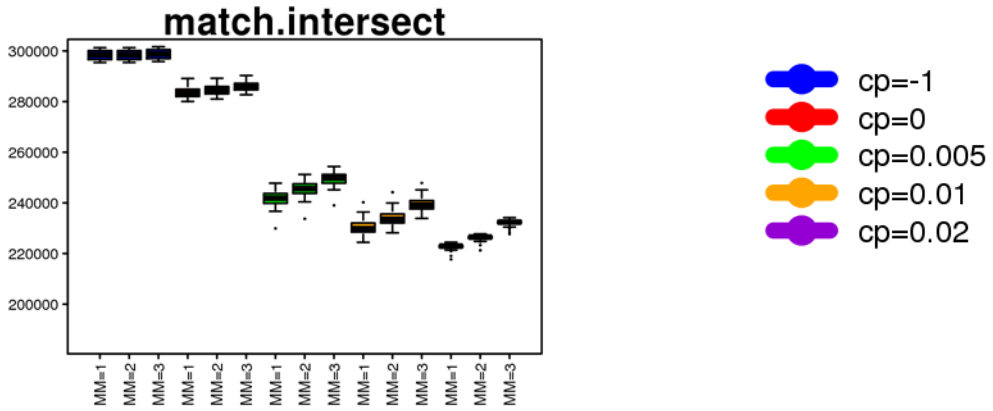


Abbildung 8.17 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Die Agentenmenge, deren Attributenkombination in der Stichprobe enthalten ist, und Observationen der realen Population entspricht, steigt mit grösser werdendem MM, wobei die Ausprägung mit steigender Stichprobengrösse abnimmt (Abbildung 8.15). Bei der richtig erzeugten Agentenmenge, deren Kombination nicht in der Stichprobe liegt, wird das Maximum bei MM=2 erreicht und das Minimum bei MM=3 (Abbildung 8.16). Betrachtet man die gesamte Menge erzeugter Agenten, denen eine Observation der realen Population zugewiesen werden kann, hängt das Maximum vom von dem cp-Wert und der Stichprobengrösse. Bei cp grösser 0 wird es bei MM=3 erreicht, bei r ∈ [0.05; 0.1] wird es auch für cp ∈ [-1; 0] bei MM=3 erreicht. Bei r=0.01 hingegen wird das Maximum bei cp ∈ [-1; 0] mit MM=2 erreicht (Abbildung 8.17).

CP BOXPLOTS:

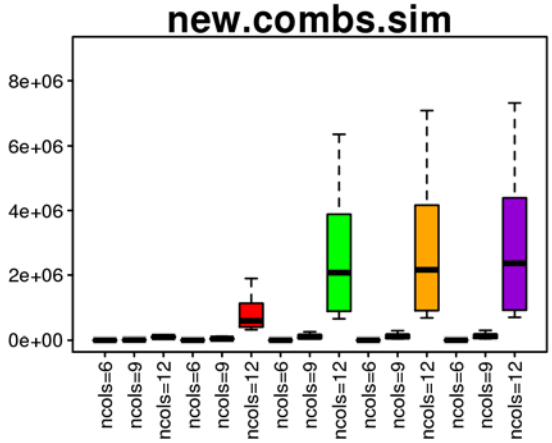


Abbildung 9.1 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

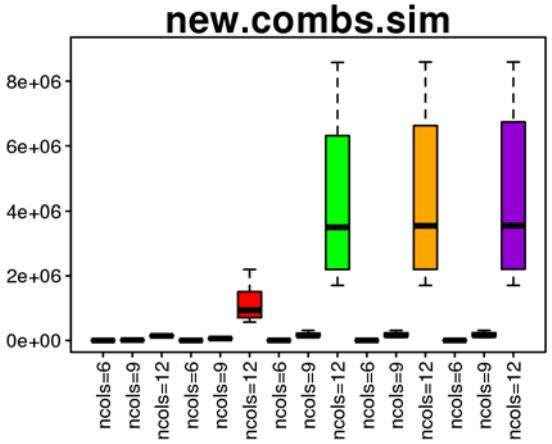


Abbildung 9.1 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

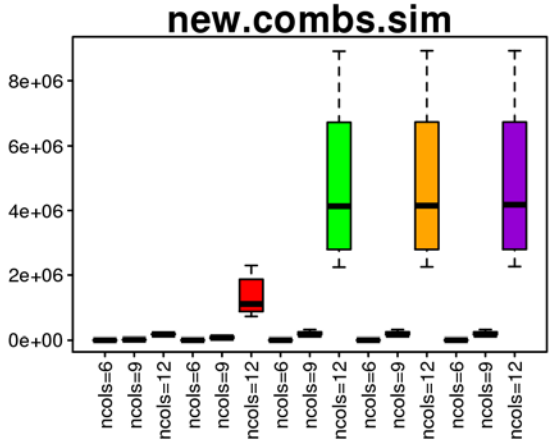
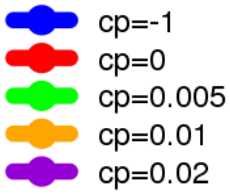


Abbildung 9.1 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



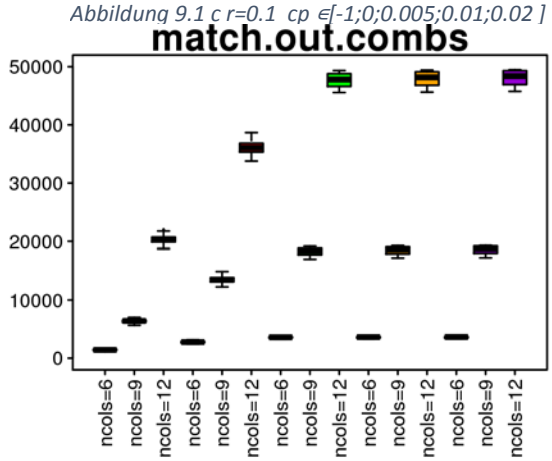


Abbildung 9.2 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

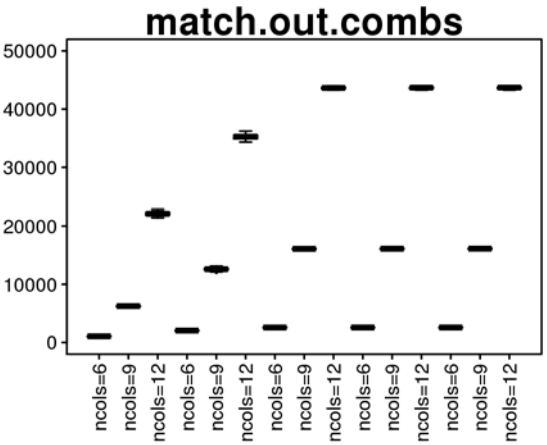


Abbildung 9.2 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

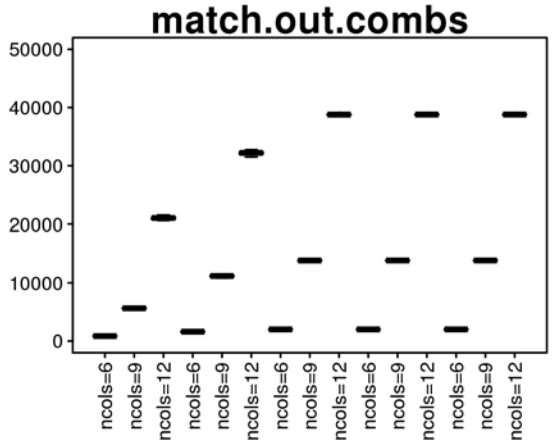
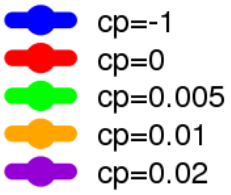


Abbildung 9.2 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



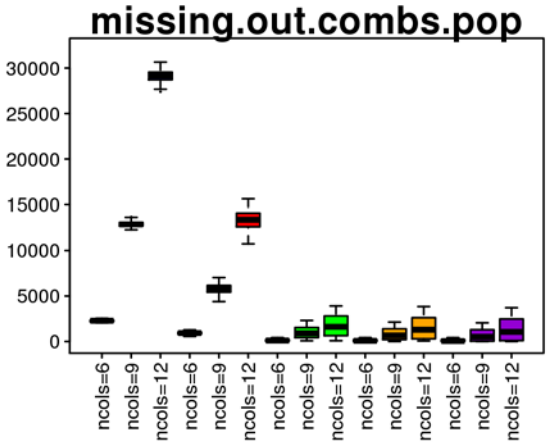


Abbildung 9.3 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

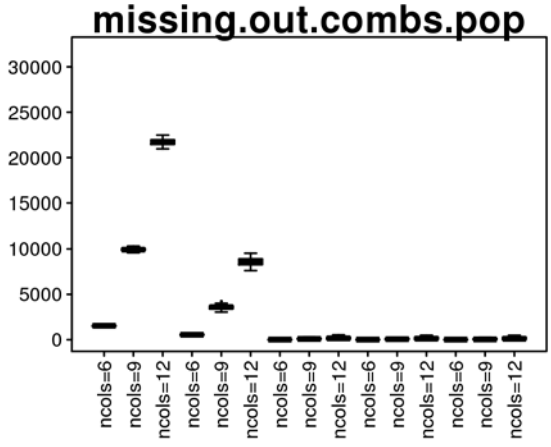
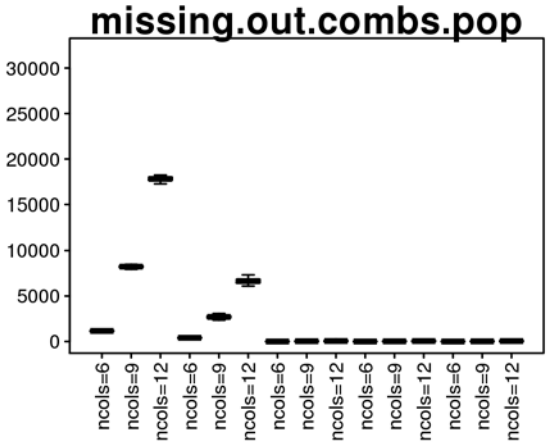


Abbildung 9.3 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



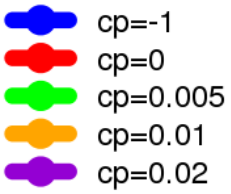


Abbildung 9.3 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Abbildung 9.1 zeigt, dass eine Erhöhung des cp-Wertes, d.h. eine grössere Abschneidung von Unterbäumen, eine Vergrösserung der Menge der neu simulierten Kombinationen zur Folge hat, wobei dieser Effekt mit steigender Attributenanzahl ausgeprägter wird. Dies resultiert aus der grösser werdenden Kombinationsmenge, die mit steigender Attributenazahl deutlich grösser wird, im Vergleich zu dem immer kleineren werdenden Kombinationsgehalt der Stichprobe. Es ist auch festzustellen, dass diese Menge bei grösseren Stichproben zunimmt. Betrachtet man die Menge der simulierten Kombinationen, die in der realen Population vorkommt, aber nicht in der Stichprobe, ist die Tendenz zu erkennen, dass diese mit stärker abgeschnittenen Bäumen grösser wird. Dabei nimmt dieses Phänomen zwischen cp ∈ [0.005; 0.01; 0.02] stark ab (Abbildung 9.2). Die Anzahl fehlender Kombinationen, die in der realen Population, aber nicht in der Stichprobe zu finden sind, nimmt mit kleineren cp-Werten ab. Auch hier ist die Differenz zwischen cp ∈ [0.005; 0.01; 0.02] sehr gering (Abbildung 9.3). Diese Menge schrumpft mir grösser werdendem r.

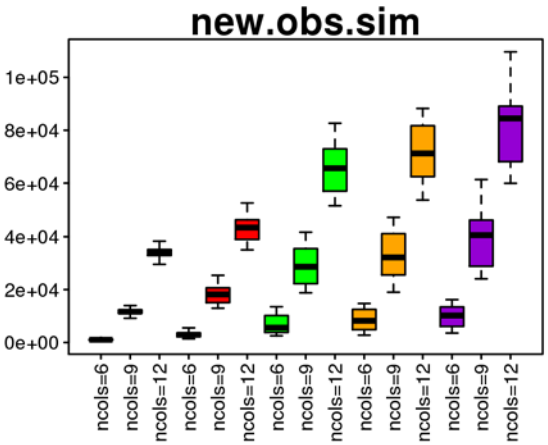


Abbildung 9.4 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

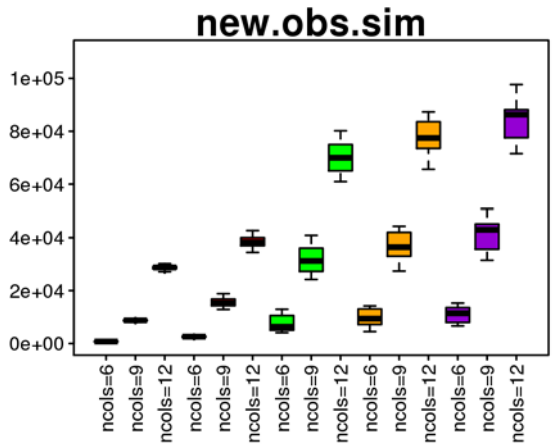
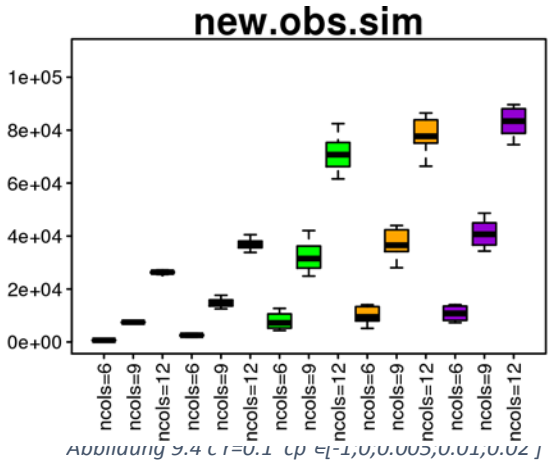


Abbildung 9.4 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



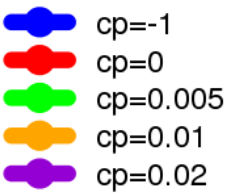


Abbildung 9.4 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

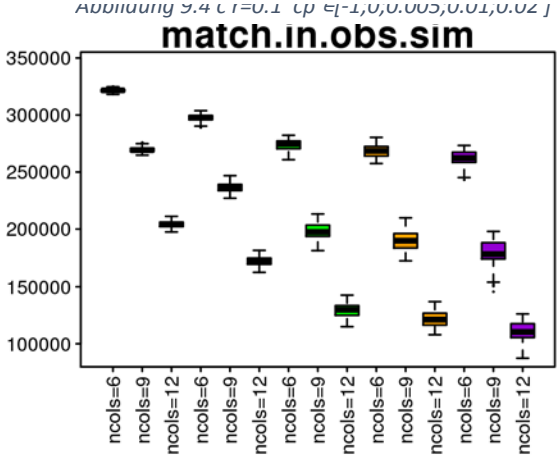


Abbildung 9.5 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

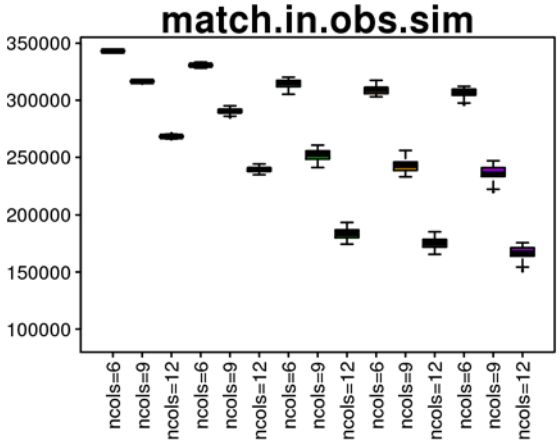
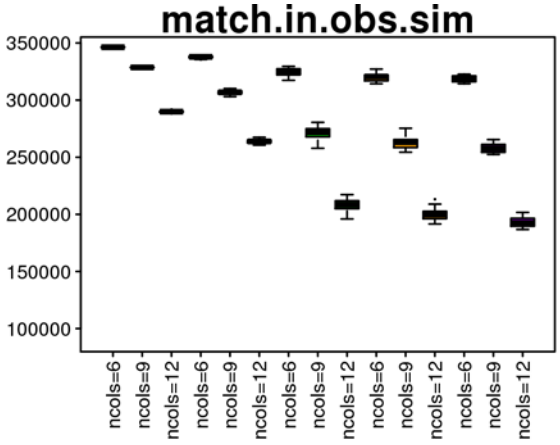


Abbildung 9.5 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]



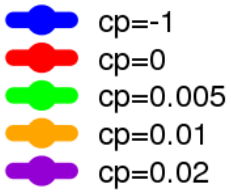


Abbildung 9.5 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Die Anzahl neu simulierter Observationen, deren Attributenkombination keiner realen Person gleicht, wächst mit steigendem cp (Abbildung 9.4). Abbildung 9.5 zeigt die Menge von erzeugten Agenten, deren Attributenkombination aus der Stichprobe stammt. Es ist deutlich zu erkennen, dass diese mit einer grösseren Abtrennung der Unterbäume abnimmt. Dies ist eine direkte Konsequenz der immer geringer werdenden Abhängigkeit des Modells von den Daten, mit denen es erstellt wird. Auch die Stichprobengrösse und die betrachtete Kombinationslänge beeinflusst die Ausprägung dieser Menge. Bei wenigen berücksichtigten Attributen ist es offensichtlich, dass die Observationen des Typs match.in den grössten Anteil in der synthetischen Population abdecken, da die meisten in der Stichprobe zu finden sind. Auch eine grösser werdende Stichprobe generiert diesen Effekt. Bei einer grösseren Attributenanzahl und/oder kleineren Stichproben wird die gleiche Überlegung gemacht und daher ist klar, dass die *match.out.obs.sim*-Menge kleiner wird. Die Anzahl erzeugter Agenten, deren Attributenkombination in der realen Population zu finden, aber nicht in der Stichprobe enthalten ist, wächst mit grösser werdendem cp-Wert (Abbildung 9.6). Auch dies ist eine Konsequenz des Anpassungsgrades des Algorithmus an die „training"-Daten. Die Anzahl dieser match.out Observationen korreliert stark mit der Stichprobengrösse und der Attributenanzahl. Dies kann wieder mittels Betrachtung des Informationsgehalts der Stichprobe im Bezug zu den Gesamtinformationen, die in realen Population enthalten sind, verstanden werden. Eine kleinere Attributenanzahl und eine grössere Stichprobe lassen dem Algorithmus weniger Spielraum bei der Erzeugung von neuen Kombinationen, da die meisten schon in der Stichprobe zu finden sind und deshalb die Anzahl neu erzeugbarer abnimmt.

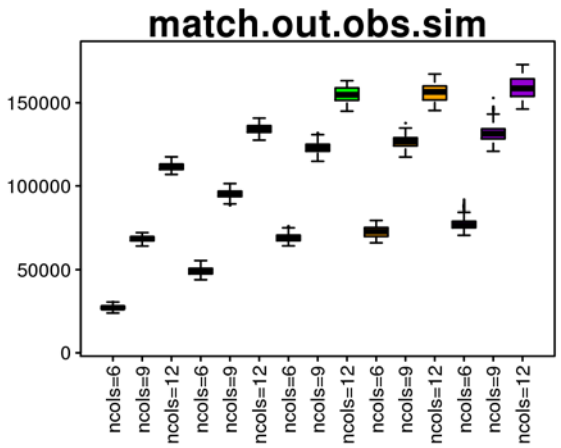


Abbildung 9.6 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

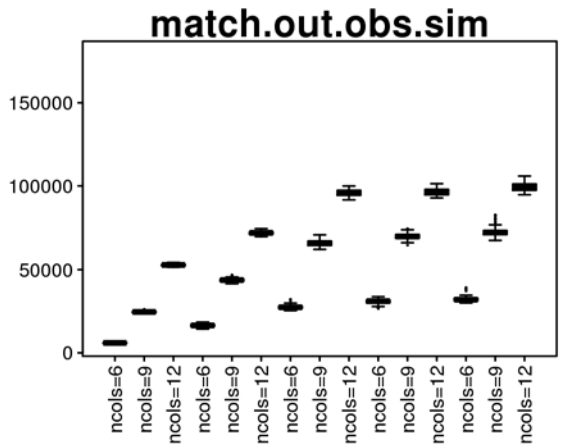


Abbildung 9.6 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

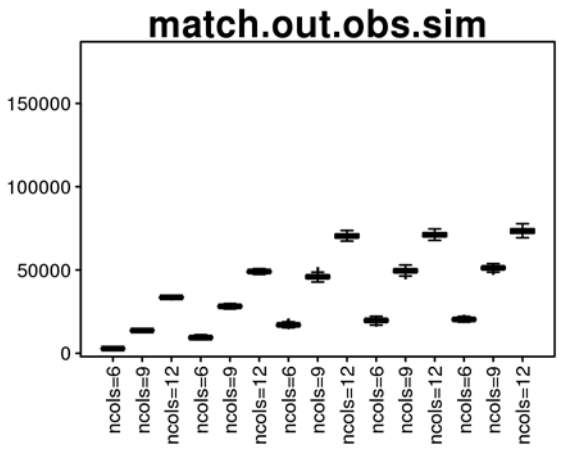
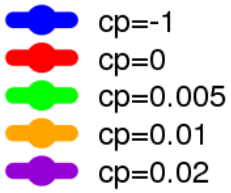


Abbildung 9.6 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



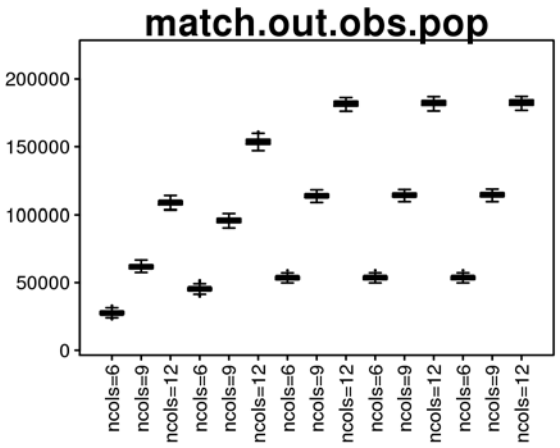


Abbildung 9.7 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

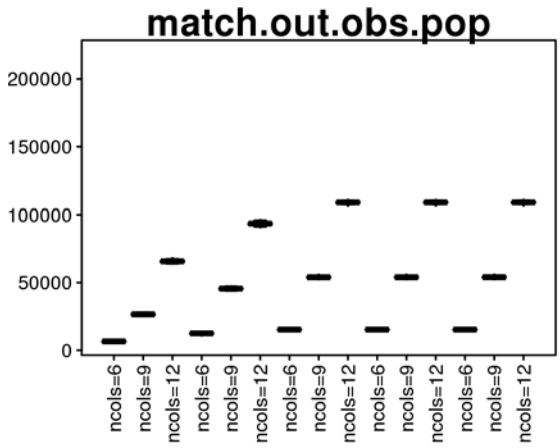


Abbildung 9.7 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

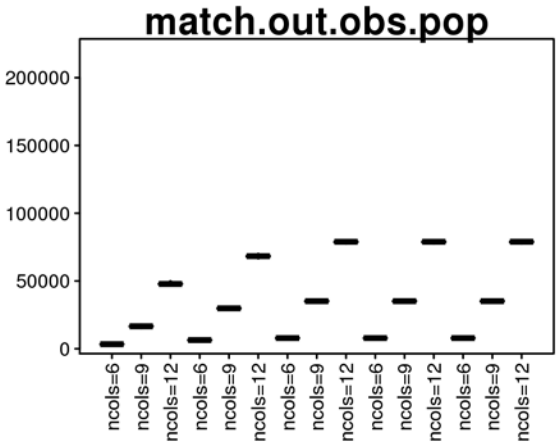
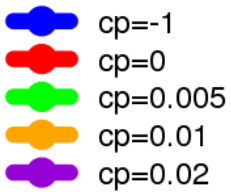


Abbildung 9.7 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



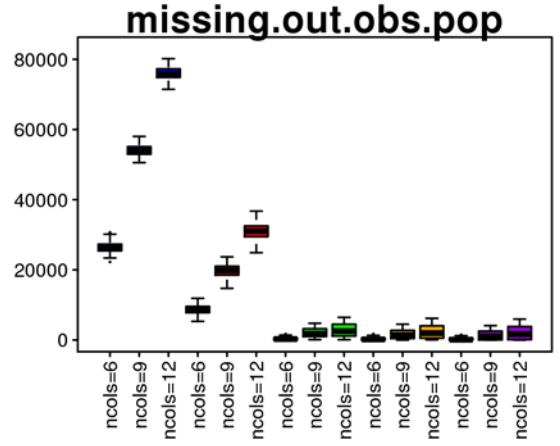


Abbildung 9.8 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

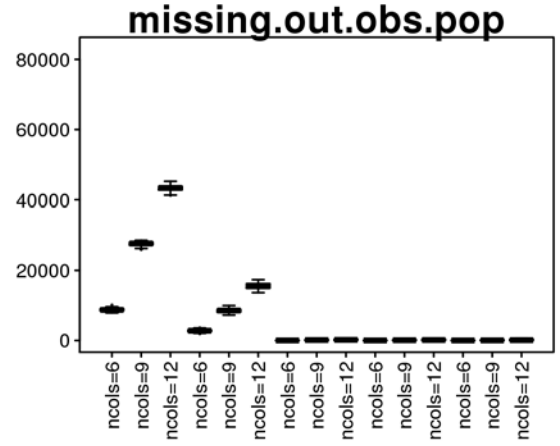


Abbildung 9.8 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

Die Observationsindikatoren, die auf die reale Population bezogen sind, bestätigen die Aussagen der simulierten Observationsindikatoren. Ein grösserer cp-Wert führt dazu, dass insgesamt mehr Agenten erzeugt werden, die einer Person der realen Population gleichen, die nicht in der Stichprobe auftaucht (Abbildung 9.7). Ein kleiner werdender cp hat zur Folge, dass mehr reale Personen, deren Attributenkombination nicht von der Stichprobe eingefangen wurde, in der synthetischen Population fehlen (Abbildung 9.8).

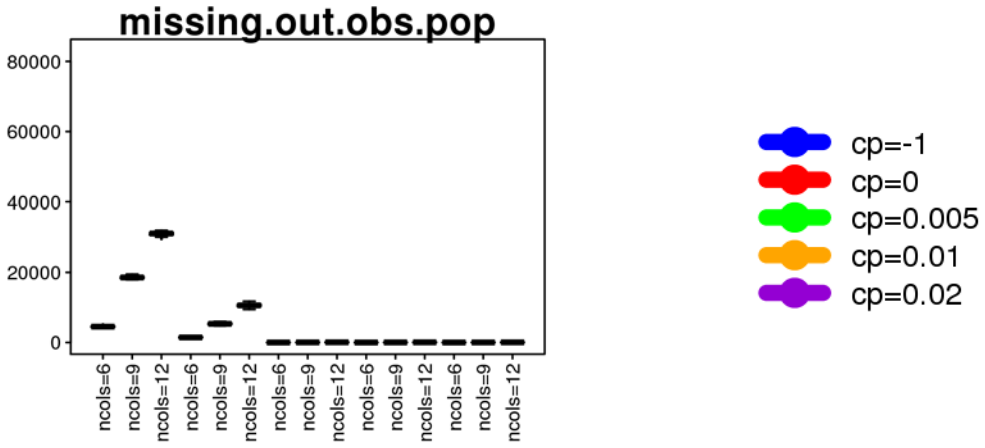


Abbildung 9.8 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

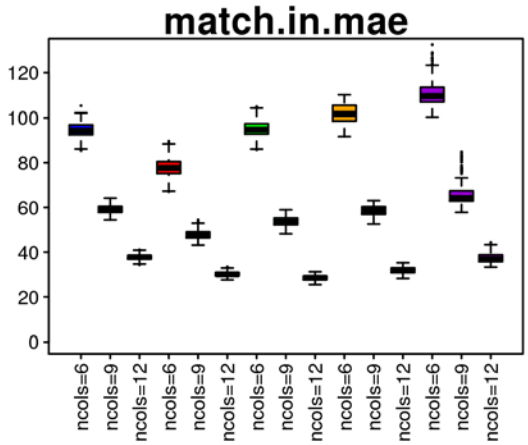


Abbildung 9.9 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

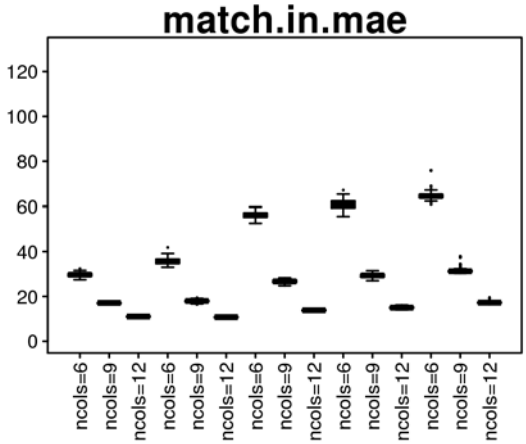


Abbildung 9.9 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

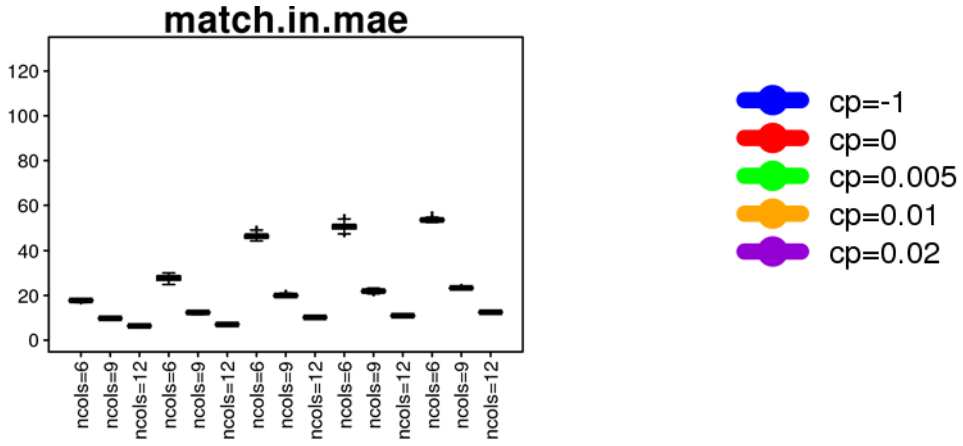


Abbildung 9.9 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

Abbildungen 9.9, 9.10, 9.11, 9.12 zeigen den absoluten und den relativen mittleren Fehler der synthetisierten Agenten des Typs match.in und match.out. Es kann kein eindeutiger, direkter Zusammenhang der Fehlerindikatoren mit cp festgemacht werden.

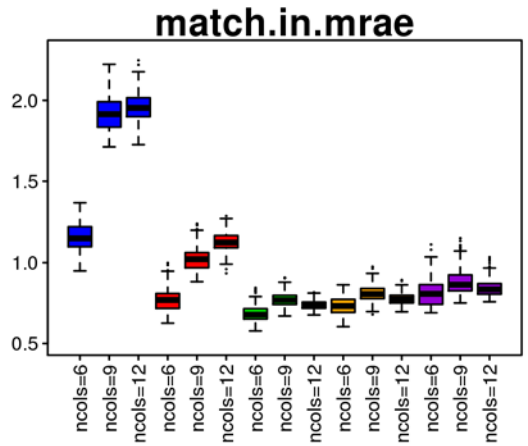


Abbildung 9.10 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

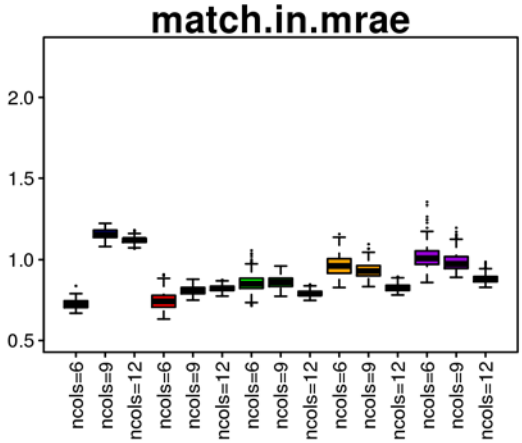


Abbildung 9.10 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

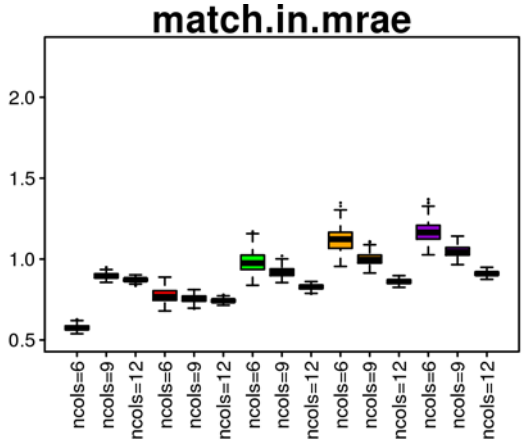
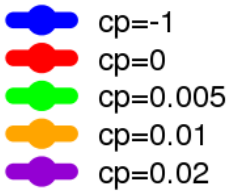


Abbildung 9.10 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



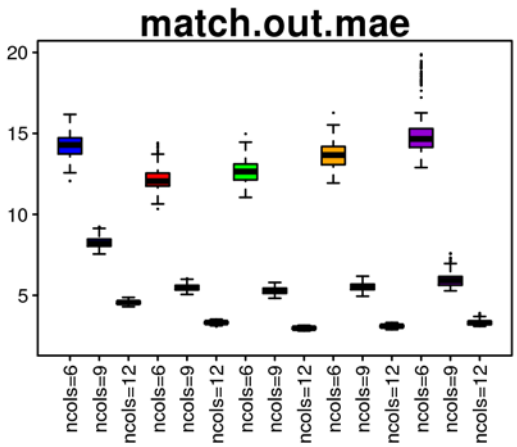


Abbildung 9.11 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

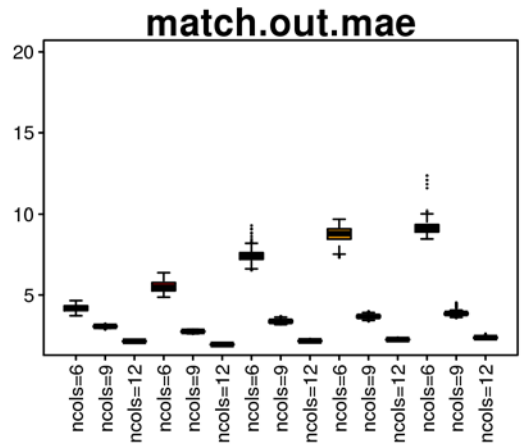


Abbildung 9.11 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

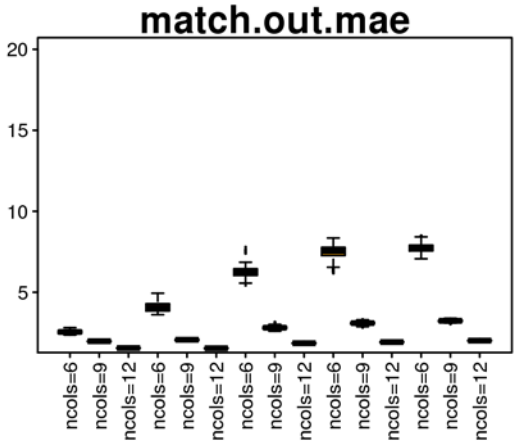
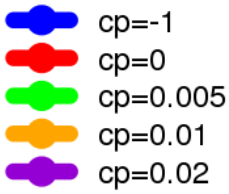


Abbildung 9.11 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



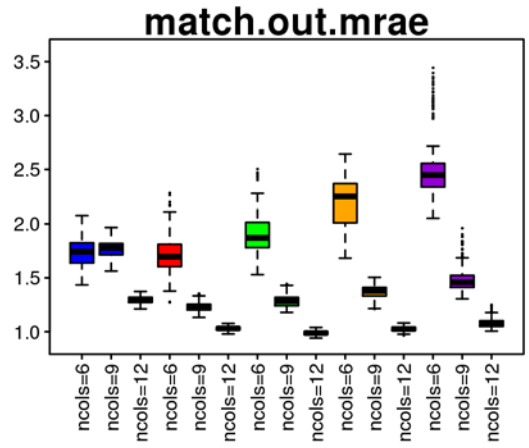


Abbildung 9.12 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

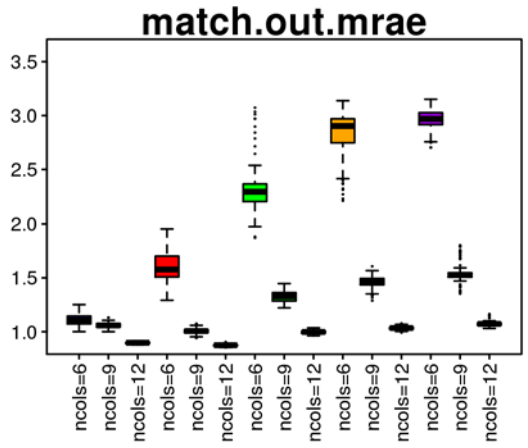


Abbildung 9.12 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

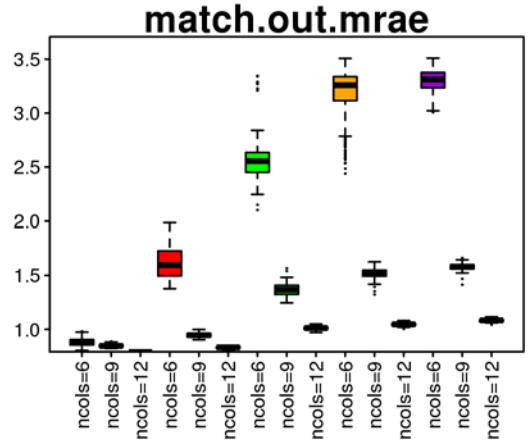
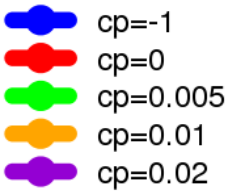


Abbildung 9.12 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



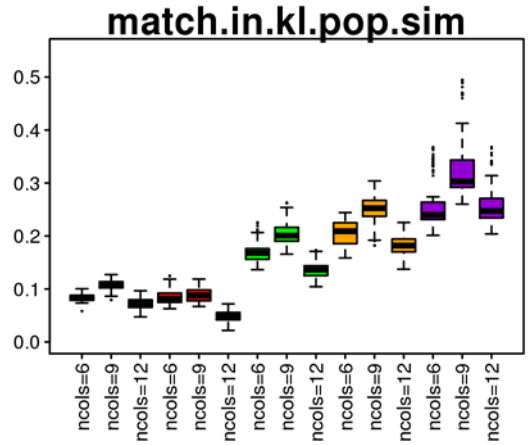


Abbildung 9.13 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

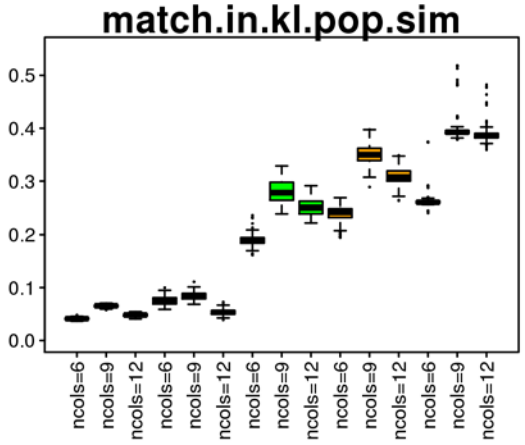


Abbildung 9.13 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

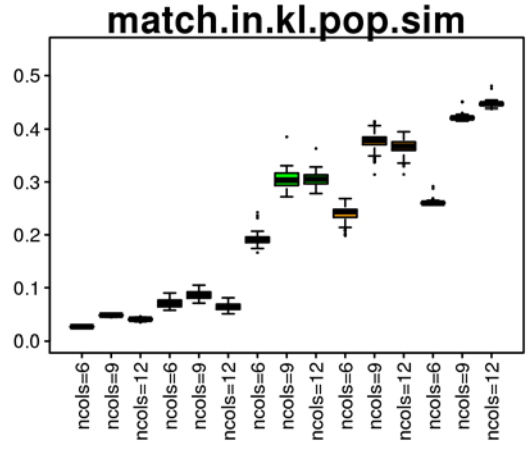
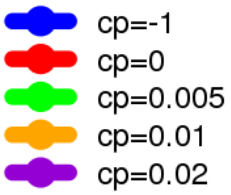


Abbildung 9.13 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



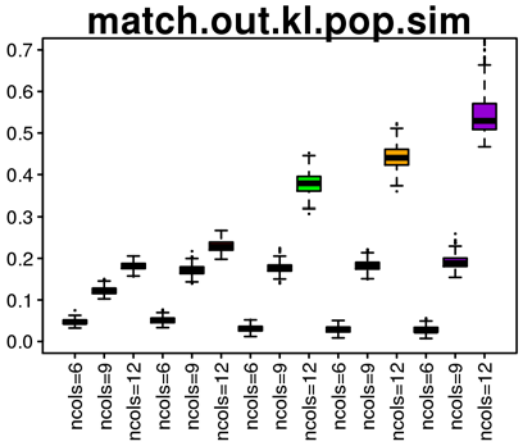


Abbildung 9.14 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

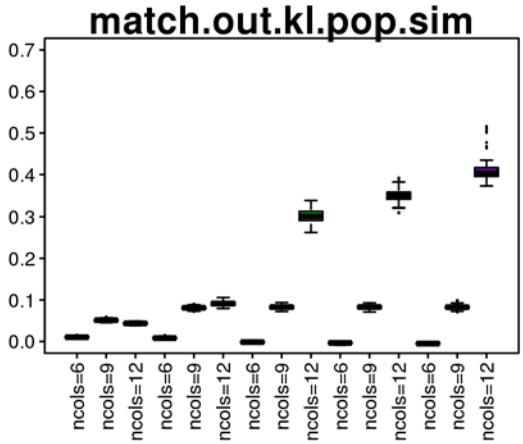


Abbildung 9.14 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

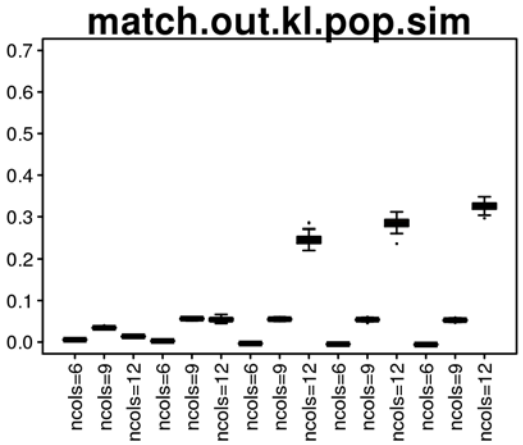
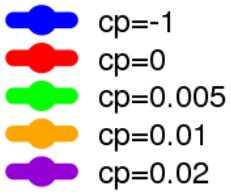


Abbildung 9.14 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



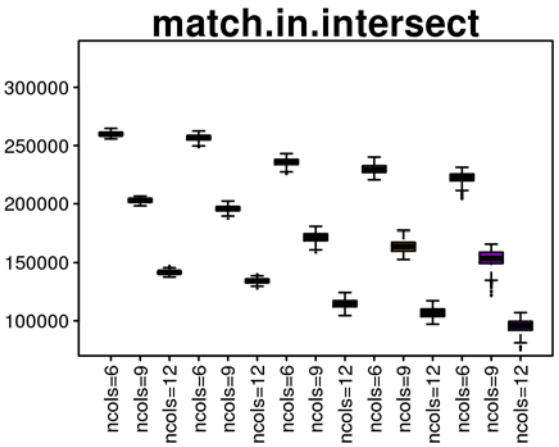


Abbildung 9.15 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

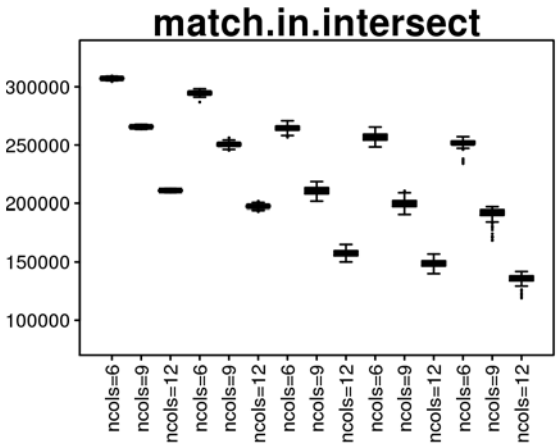


Abbildung 9.15 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

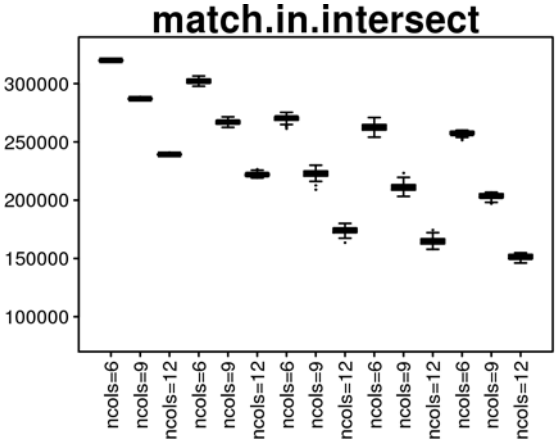


Abbildung 9.15 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]

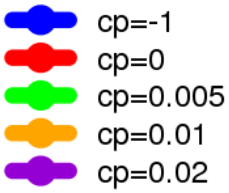


Abbildung 9.15 zeigt, dass *match.in.intersect* bei den maximalen Regressionsbäumen für jedes Szenario am grössten ausfällt. Betrachtet man den Indikator *match.out.intersect*, ist die entgegengesetzte Tendenz zu erkennen. Aus einer Erhöhung des cp-Wertes im Intervall [-1; 0; 0.005] resultiert eine grössere Durchschnittsmenge zwischen der realen und der simulierten Population, für die Agenten deren Attributenkombination nicht aus der Stichprobe stammt. Überschreitet man die Schwelle von cp=0.005, wird diese Menge wieder kleiner (Abbildung 9.16). Abbildung 9.17 zeigt die Ausprägung der gesamten Durchschnittsmenge zwischen der realen und der simulierten Population in Funktion von cp. Bei r=0.01 ist für alle Attributenanzahlen das gleiche Muster erkennbar. Der grösste Wert für match.intersect wird bei cp=0 erreicht und er wird bei einer Erhöhung von cp kleiner. Für die grösseren Stichproben hingegen ist diese Menge bei cp=-1 maximal und wird mit einer Erhöhung von cp immer kleiner.

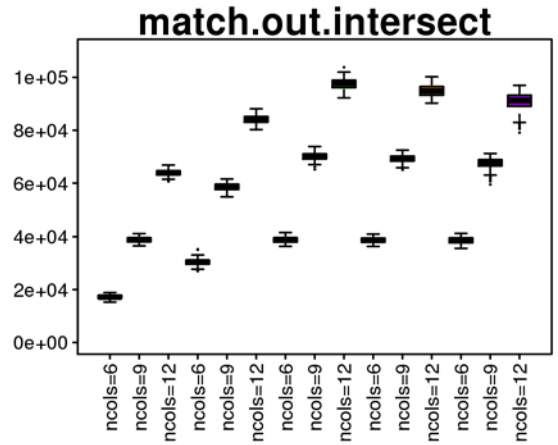


Abbildung 9.16 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

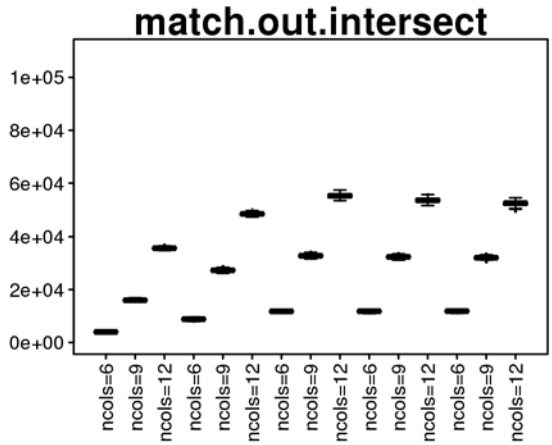


Abbildung 9.16 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

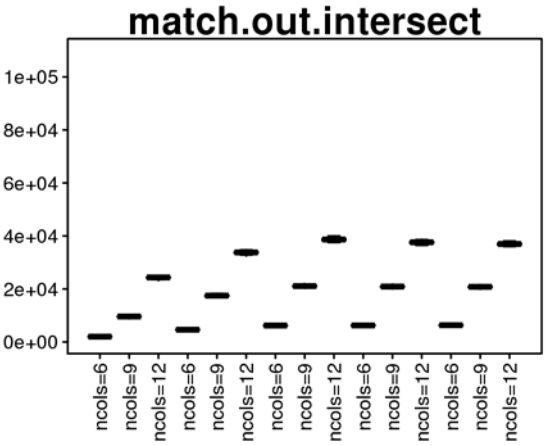
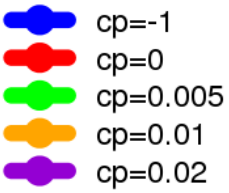


Abbildung 9.16 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]



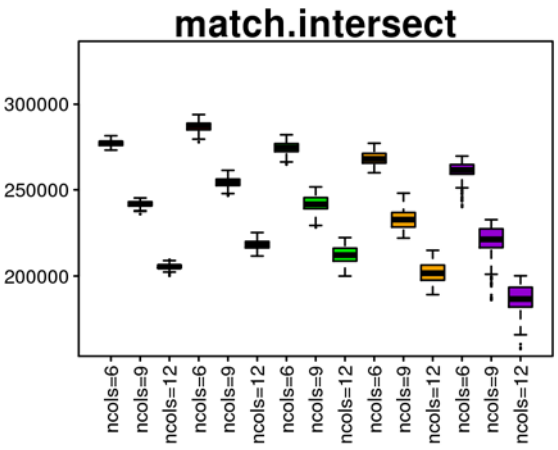


Abbildung 9.17 a r=0.01 cp ∈[-1;0;0.005;0.01;0.02]

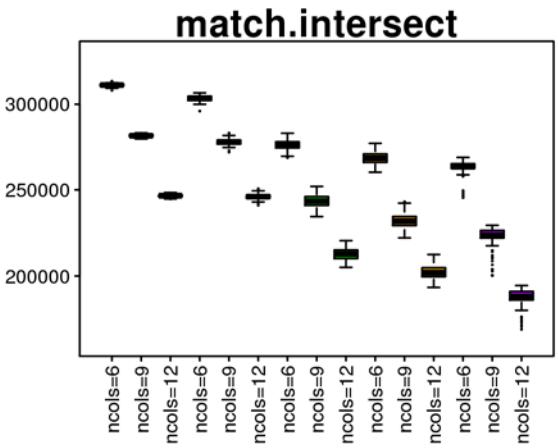


Abbildung 9.17 b r=0.05 cp ∈[-1;0;0.005;0.01;0.02]

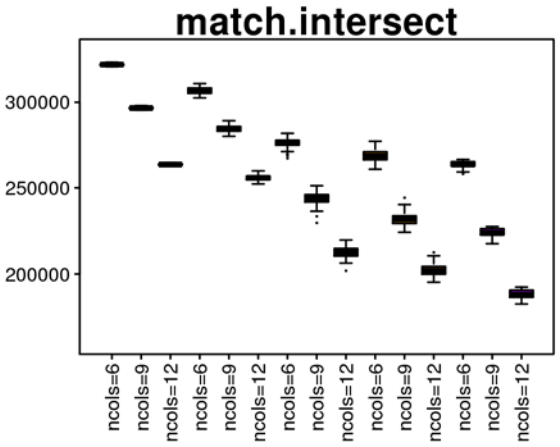


Abbildung 9.17 c r=0.1 cp ∈[-1;0;0.005;0.01;0.02]