„Kleine Fische, große Fische“ oder „Selektion simulieren“

Tobias Kellner, Dr. Hubert Langlotz, Dr. Wilfried Zappe[1]

Lebewesen, die länger leben, können sich häufiger fortpflanzen und somit ihre Gene öfter weiter vererben. Eine längere Lebensdauer hängt aber meistens davon ab, wie gut die Lebewesen an ihre Umwelt angepasst sind. Natürlich pflanzen sich auch weniger gut angepasste Lebewesen fort, aber eben nicht so häufig wie die besser akklimatisierten. Es kommt im Laufe der Zeit dadurch statistisch gesehen zu einem Übergewicht der besser angepassten Individuen. Man nennt diesen Vorgang „natürliche Selektion“. Auf diese Weise werden günstige Allele im Genpool häufiger, nachteilige Allele seltener[2].

Ein gut verständliches Beispiel ist die Entwicklung der Körpergröße von Kabeljau im Nordost-Atlantik. Durch die Überfischung entstand ein großer Selektionsdruck. Kleinere Fische hatten bessere Chancen durch die engmaschigen Netze zu entkommen als große Fische. So waren Kabeljau aus dem Nordost-Atlantik vor 60 Jahren, als die Jagd auf sie begann, im Schnitt noch 95 cm groß, heute erreichen sie nur noch eine Körpergröße von 65 cm[3].

Ein einfaches mathematisches Modell

Eine Fischsorte sei durch A-Kugeln, die andere Sorte durch B-Kugeln repräsentiert, die äußerlich völlig gleich aussehen und sich gleich anfühlen, aber sich z. B. durch einen Aufdruck „A“ bzw. „B“ oder in ihrer Farbe unterscheiden. Aus einer großen Anzahl von A-Kugeln und B-Kugeln denken wir uns zehn A-und zehn B-Kugeln ausgewählt und in eine Urne gelegt. Aus dieser Urne werden „auf gut Glück“ mit einem Griff zehn Kugeln entnommen und beiseite gelegt. Sie stehen für gestorbene Fische. Nun sind noch zehn Kugeln in der Urne. Zu jeder in der Urne verbliebenen A-Kugel wird eine weitere A-Kugel und zu jeder verbliebenen B-Kugel eine weitere B-Kugel hinzugefügt, so dass sich wieder 20 Kugeln insgesamt in der Urne befinden. Dieser Vorgang kann als Simulation der Fortpflanzung der Fische aufgefasst werden. Nun werden wieder zufällig zehn Kugeln entnommen und es wird wieder so verfahren, wie oben beschrieben.

„Halbschriftliche“ Simulationen mit dem TI-Nspire™ CAS

Auf einer Notes-Seite wird die folgende Ausgangssituation geschaffen:

Abb. 1

1) Es wurden also im ersten Durchgang sechs A-Kugeln gezogen, es verbleiben noch vier in der Urne.
2) Diese verdoppelt ergeben 8 A-Kugeln (anz_a) und analog 12 B-Kugeln (anz_b) als neue Urnenfüllung.
3) Diese Zahlen sind in der ersten und zweiten Zeile als neue Endzahlen der seq-Befehle einzutragen und durch ENTER zu bestätigen.
4) Liste *list2* wird durch Anklicken neu aktiviert:

Abb. 2

Die Schritte (1) bis (4) werden nun mit den neuen Werten der Variablen *anz_a* und *anz_b* wiederholt usw., bis eine dieser Variablen den Wert 0 hat.

Die Ergebnisse werden handschriftlich in einer Tabelle notiert. Das Zufallsexperiment wird nach der im Abschnitt „mathematisches Modell“ beschriebenen Methode solange wiederholt, bis nur noch Kugeln einer Sorte in der Urne sind. Die Tabelle zeigt ein mögliches Ergebnis:

Runde	In der Urne sind		Gezogen wurden		Nach dem Ziehen bleiben	
	A-Kugeln	B-Kugeln	A-Kugeln	B-Kugeln	A-Kugeln	B-Kugeln
0	10	10	6	4	4	6
1	8	12	2	8	6	4
2	12	8	10	0	2	8
3	4	16	3	7	1	9
4	2	18	1	9	1	9
5	2	18	1	9	1	9
6	2	18	2	8	0	10

Es ist auch bei häufiger Wiederholung dieser Versuchsreihe gut zu sehen, dass stets genau eine Sorte von Kugeln übrigbleibt. Wenn eine Sorte Kugeln das „Übergewicht“ erhält, wird die Wahrscheinlichkeit größer, beim nächsten Ziehen auch mehr Kugeln aus dieser Sorte zu erhalten. Das ist aber keinesfalls sicher, so dass im Laufe des Prozesses auch das Übergewicht zur anderen Sorte wechseln kann. Dieses Modell einer Selektion lässt sich am besten in Gruppenarbeit realisieren.

Das Zufallsexperiment lässt sich auch mit anderen Startzahlen durchführen, z. B. mit fünf A- und fünf B-Kugeln zu Beginn und dem Ziehen von jeweils sechs Kugeln mit einem Griff. Hier wird man i. A. weniger Ziehungen brauchen, um eine vollständige Selektion einer Kugelsorte zu erreichen.

Simulation mithilfe eines Ein-Zeilen-Programms

Wir starten mit der Festlegung der Liste x in der Form

x:={a,a,a,a,a,a,a,a,a,a,b,b,b,b,b,b,b,b,b,b}.

1) Durch zufälliges Umsortieren mit *randsamp(x, 20, 1)* der Liste x entsteht eine Liste y.
2) Aus der Liste y werden mit *seq(y[k], k, 1, 10)* die ersten zehn Elemente ausgewählt. Die verbleibenden zehn Kugeln können als die ausgesonderten betrachtet werden, die zehn ausgewählten Kugeln gelten dann als die in der Urne verbliebenen Kugeln. Aus dieser neuen Liste und ihrer Verdopplung mit *augment(seq(y[k], k, 1, 10), seq(y[k], k, 1, 10))* entsteht eine neue Liste x. (Das entspricht der neuen Urnenfüllung mit der verdoppelten Anzahl der Restkugeln.)
3) Beide Anweisungen werden in ein und derselben Zeile durch einen Doppelpunkt getrennt. Fertig ist das Ein-Zeilen-Programm!
4) Jetzt kann es wieder mit Schritt (1) losgehen.

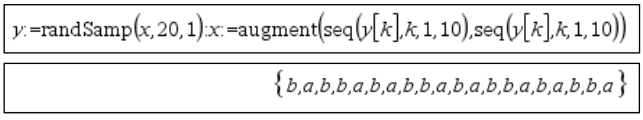

Abb. 3a (oben): Eingabezeile, Abb. 3b (unten): Ausgabezeile

Nun muss man nur noch wiederholt ENTER drücken, um jedes Mal eine neue Realisierung des Ein-Zeilen-Programms zu erreichen. Dies macht man solange, bis nur noch eine Sorte der Buchstaben a bzw. b übrig bleibt.

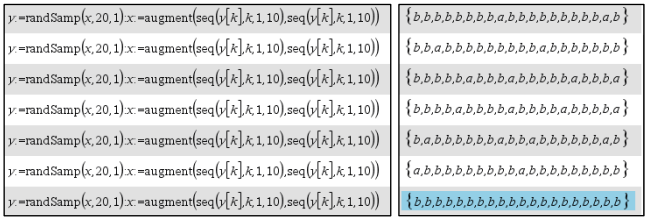

Abb. 4

Simulation mithilfe einer Funktion

Für diesen Ansatz wird eine Funktion wie folgt definiert:

$$f : R^3 \rightarrow R^2, f(w,s,i) = 2\begin{bmatrix} w-i \\ s-10+i \end{bmatrix}$$

Dabei beschreibt w die Anzahl der A-Kugeln, s die Anzahl der B Kugeln und i die Anzahl der gezogenen A-Kugeln. Die Funktion zieht nun die gezogenen Kugeln (i A-Kugeln und 10-i B-Kugeln) von der jeweiligen vorherigen Anzahl w bzw. s ab, verdoppelt die Ergebnisse und speichert sie in einen Vektor.

Um daraus wiederum ein Ein-Zeilen-Programm zu erstellen, erstellen wir zunächst einen Startvektor v, der den ersten Zug simuliert. Dazu wird v mit f(10,10,randInt(0,10)) initialisiert, also 10 A-Kugeln, 10 B-Kugeln und einer zufälligen Anzahl an gezogenen A-Kugeln zwischen 0 und 10.

Nun wird jeder neue Schritt dadurch simuliert, dass f erneut mit den Werten von v und einer neuen zufällig gezogenen Anzahl an A-Kugeln aufgerufen und das Ergebnis wieder in v gespeichert wird.

Dabei gilt für die einzelnen Definitionswerte:

- w muss die letzte Anzahl der A-Kugeln sein, also der erste Eintrag in v, hier: norm(v[1])
- s muss die letzte Anzahl der B-Kugeln sein, also der erste Eintrag in v, hier: norm(v[2])
- für i, die neue Anzahl an gezogenen A-Kugeln, müssen die oberen und unteren Grenzen angepasst werden:
  Es können nie mehr A-Kugeln als vorhanden gezogen werden und auch nie mehr als 10, also ist die obere Grenze min({10,norm(v[1])}).
  Es können nie weniger als 0 A-Kugeln gezogen werden aber wenn weniger als 10 B-Kugeln vorhanden sind, muss der Rest der gezogenen Kugeln als A-Kugeln gezogen werden. Das sind also 10 weniger der Anzahl der B-Kugeln. Damit ist die untere Grenze 10-min({10,norm(v[2])}).

Somit ergibt sich als neues Ein-Zeilen-Programm (siehe Abbildung 5)

v := f(norm(v[1]), norm(v[2]),randInt(10-min({10,norm(v[2])}), min({10,norm(v[1])})))

und wie im oben beschriebenen Programm lässt sich die Simulation durch wiederholtes Drücken der ENTER-Taste durchführen.

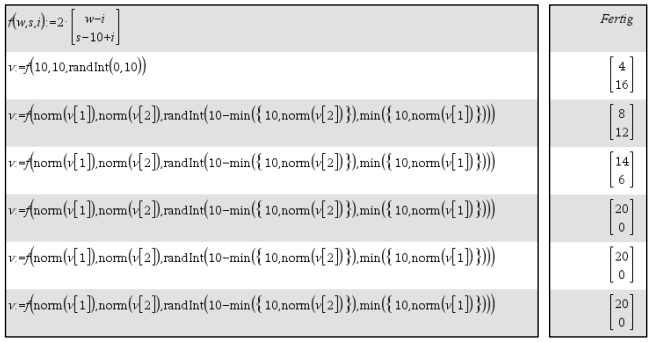

Abb. 5

Simulation mithilfe eines Programms und einer Funktion

Dem in Abbildung 6 dargestellten Programm liegen folgende Überlegungen zugrunde: Die Liste li wird zunächst erzeugt durch zwei Listen, die n-mal die Ziffern 0 bzw. 1 beinhalten. Die Liste li3 beinhaltet die gezogenen Elemente. Um die Listen li1 und li2 und damit die Ausgangsliste neu zu belegen, wird noch eine Hilfsliste li4 benötigt. In der Variable „Anzahl“ werden die Durchläufe gespeichert.

Die while-Schleife wird dann durchlaufen, solange die Summe von li größer als 0 und gleichzeitig kleiner als 2n ist.

```
Define ldrehend(n)=
Prgm
Local li1,li2,li3,li,li4,s,gs,anz
li1:=seq(0,k,1,n)
li2:=seq(1,k,1,n)
li:=augment(li1,li2)
gs:=n
anz:=0
While sum(li)>0 and sum(li)<2·n
 li3:=randSamp(li,n,1)
s:=sum(li3)
li1:=seq(0,k,1,n−(gs−s))
li2:=seq(1,k,1,gs−s)
 li4:=augment(li1,li2)
 li:=augment(li4,li4)
 gs:=sum(li)
 anz:=anz+1
 Disp li
EndWhile
Disp anz
EndPrgm
```

Abb. 6

Der Name des Programms rührt von dem Umstand her, dass Rudolf Taschner sich in seinem Buch auf linksdrehende Aminosäuren bezieht (vgl. auch Online-Ausgabe dieses Artikels in der Material-Datenbank).

Wie in der Abbildung 7 zu sehen ist, scheint die Zahl der Durchläufe stark zu schwanken.

```
ldrehend(10)
{0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,1,1,1}
{0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1}
{0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1}
{0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1}
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
5
Fertig
ldrehend(10)
{0,0,0,0,1,1,1,1,1,1,0,0,0,0,1,1,1,1,1,1}
{0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1}
{0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1}
{0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1}
{0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1}
{0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1}
{0,0,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1}
{0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1}
{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
9
Fertig
```

Abb. 7

Um hierfür einen Näherungswert für die mittlere Anzahl der Durchläufe zu bekommen, nutzt man eine Funktion, die den gleichen Aufbau wie das Programm hat, als Ausgabe aber nur die Anzahl der Durchläufe zurückgibt.

$\frac{\sum_{k=0}^{100}(linksdreh(10))}{100.}$	22.85
$\frac{\sum_{k=0}^{100}(linksdreh(10))}{100.}$	24.72
$\frac{\sum_{k=0}^{1000}(linksdreh(10))}{1000.}$	23.974
$\frac{\sum_{k=0}^{1000}(linksdreh(10))}{1000.}$	23.655

Abb. 8

Der Mittelwert scheint sich bei 24 zu stabilisieren. Eine theoretische Berechnung dieser mittleren Wartezeit kann mit Hilfe einer sogenannten absorbierenden Markov-Kette erfolgen. Mehr dazu findet man z.B. in dem hervorragenden Buch von Arthur Engel.

In seinem ibook „Intuition und Zufall“ beschreibt Benno Grabinger[5] (S. 71ff.) ein ähnliches Beispiel zur Selektion, welches vom Nobelpreisträger Manfred Eigen stammt.

Literatur, Quellen:

1) Angeregt durch ein Beispiel aus dem Buch von Rudolf Taschner: *Zahl Zeit Zufall. Alles Erfindung?*; Ecowin Verlag, 2007
2) Quelle: (Stand vom 07.10.2014) http://www.biologie-schule.de/evolutionsfaktor-selektion.php
3) Quelle: (Stand vom 07.10.2014) http://de.wikipedia.org/wiki/Selektion_%28Evolution%29
4) Arthur Engel: *Wahrscheinlichkeitsrechnung und Statistik*; Bd.2, Klett, 1992
5) Benno Grabinger: *Intuition und Zufall*; ibook

Autoren

Tobias Kellner, Dr. Hubert Langlotz, Dr. Wilfried Zappe,
Thüringen (D)