§1 Das Phänomen Zufall

Im täglichen Leben werden wir oft mit Vorgängen konfrontiert, bei denen der Zufall eine Rolle spielt. Bereits als Kind lernt man die Tücken des Zufalls kennen, wenn man beim Spiel "Mensch Ärgere Dich Nicht" sehr lange keinen Sechser würfelt. Kartenspiele sind typische Beispiele für Vorgänge, bei denen der Zufall seine Hand im Spiel hat. Glücksspiele wie Pokern, Lotto und Roulett "leben" vom Zufall und der Tatsache, dass die Spieler im allgemeinen zu wenig über die damit verbundenen zufälligen Vorgänge Bescheid wissen - sie würden sonst ein derartig hohes Risiko niemals in Kauf nehmen.

Ziel dieses Abschnitts ist es, durch Simulation von einfachen Zufallsexperimenten wie Würfeln, Münzwerfen und Ziehen von Kugeln aus einer Urne dem Leser spielerisch ein Gefühl für das Phänomen Zufall zu vermitteln.

1.1 Pseudozufallszahlen

Zahlreich durchgeführte Spiele in der Kindheit haben uns ein gewisses intuitives Gefühl für den Zufall vermittelt. Allerdings ist dieses so erworbene intuitive Gefühl oft recht trügerisch. So können wir kaum abschätzen, ob ein Würfel fair ist, also die Zahlen 1, 2, 3, 4, 5, 6 mit gleicher Wahrscheinlichkeit gewürfelt werden. Noch viel größere Probleme haben wir mit "seltenen Ereignissen". Wer käme schon auf die Idee, beim Lotto "6 aus 45" die Zahlen 1, 2, 3, 4, 5, 6 anzukreuzen (obwohl diese sechs Zahlen mit der gleichen Wahrscheinlichkeit gezogen werden, wie jede andere Kombination von sechs Zahlen).

Um dieses intuitive Gefühl für den Zufall zu vertiefen, befassen wir uns in diesem Kapitel spielerisch mit dem Würfeln, dem Werfen von Münzen und dem Ziehen von Kugeln aus einer Urne. Dabei werden wir aber nicht tatsächlich mit einem Würfel würfeln bzw Münzen werfen bzw Kugeln aus einer Urne ziehen, sondern diese zufälligen Vorgänge auf dem Computer simulieren. Die durch Simulation erzeugten Realisierungen verhalten sich dabei in jeder Hinsicht ebenso wie jene Realisierungen, die durch tatsächliches Experimentieren gewonnen werden.

Zentral für dieses Simulieren von zufälligen Vorgängen sind die in *Mathematica* implementierten Befehle

■ RandomReal[] bzw RandomReal[{zmin, zmax}]

liefert eine auf dem Intervall [0, 1] bzw dem Intervall [zmin, zmax] gleichmäßig verteilte Pseudozufallszahl.

■ RandomReal[{zmin, zmax}, *k*]

erzeugt eine Liste von *k* auf dem Intervall [zmin, zmax] gleichmäßig verteilten Pseudozufallszahlen.

■ RandomInteger[] bzw RandomInteger[{zmin, zmax}]

liefert eine auf der Menge {0, 1} bzw der Menge $\{i \mid \text{zmin} \leq i \leq \text{zmax}\}$ gleichmäßig verteilte Pseudozufallszahl.

■ RandomInteger[{zmin, zmax}, *k*]

erzeugt eine Liste von *k* auf der Menge $\{i \mid \text{zmin} \leq i \leq \text{zmax}\}$ gleichmäßig verteilten Pseudozufallszahlen.

■ RandomSample[*liste*]

liefert eine pseudozufällige Permutation der Liste *liste*.

■ RandomSample[*liste*, *k*]

erzeugt eine Liste mit den ersten *k* Elementen einer pseudozufälligen Permutation der Liste *liste*.

▼

Beispielsweise gilt

```
RandomReal[{2, 3}]
```

```
2.1597
```

```
RandomReal[{2, 3}, 5]
```

```
{2.32626, 2.17111, 2.97234, 2.49412, 2.90823}
```

```
RandomInteger[{1, 6}]
```

```
6
```

```
RandomInteger[{1, 6}, 5]
```

```
{1, 2, 1, 6, 4}
```

```
RandomSample[{a, b, a, c, a, b, c}]
```

```
{c, b, a, b, c, a, a}
```

```
RandomSample[{a, b, a, c, a, b, c}, 5]
```

```
{b, c, a, a, b}
```

1.2 Würfeln

Würfelspiele gehören zu jenen Spielen, mit denen man bereits als Kind intensiv Bekanntschaft macht. Wir wollen uns daher als erstes mit dem Würfeln näher befassen. Dabei setzen wir voraus, dass wir es mit einem homogenen Würfel, also einem Würfel, der die Augen 1, 2, 3, 4, 5, 6 mit der gleichen Wahrscheinlichkeit zeigt, zu tun haben.

1.2.1 Beispiel: Unser Zufallsexperiment besteht im einmaligen Werfen eines Würfels.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Wir simulieren das einmalige Werfen eines Würfels mit Hilfe von RandomInteger:

```
RandomInteger[{1, 6}]
```

```
6
```

1.2.2 Beispiel: Unser Zufallsexperiment besteht im einmaligen Werfen von *k* Würfeln.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Wir simulieren das einmalige Werfen von *k* Würfeln mit Hilfe von RandomInteger (die auf diese Weise erzeugte Realisierung gibt an, mit welchem Würfel welche Zahl geworfen wurde):

```
k = 10;
RandomInteger[{1, 6}, k]
Clear[k]
```

```
{3, 6, 2, 1, 1, 2, 4, 2, 2, 3}
```

1.2.3 Beispiel: Unser Zufallsexperiment besteht im *m*-maligen Werfen von *k* Würfeln.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Analog zu Beispiel 1.2.2 simulieren wir mit Hilfe von RandomInteger das einmalige Werfen von *k* Würfeln. Diesen Vorgang wiederholen wir mit Hilfe von Table *m* mal (die damit erzeugte Realisierung gibt die bei den einzelnen Wiederholungen gewürfelten Augenzahlen der *k* Würfel an):

```
k = 5; m = 10;
Table[RandomInteger[{1, 6}, k], {m}]
Clear[k, m]
```

```
{{1, 1, 5, 5, 6}, {4, 5, 3, 3, 3}, {5, 2, 3, 6, 1}, {3, 1, 1, 3, 3}, {6, 6, 2, 3, 1},
 {3, 5, 5, 3, 3}, {3, 6, 6, 5, 1}, {2, 6, 6, 3, 2}, {1, 1, 3, 6, 2}, {6, 5, 5, 1, 4}}
```

1.2.4 Beispiel: Unser Zufallsexperiment besteht im *m*-maligen Werfen von *k* Würfeln und der anschließenden Beobachtung der jeweils geworfenen Augensumme.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte wieder, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Analog zu Beispiel 1.2.2 simulieren wir zuerst das einmalige Werfen von *k* Würfeln und ermitteln anschließend mit Hilfe von Apply und Plus die dabei gewürfelte Augensumme. Diesen Vorgang wiederholen wir mit Hilfe von Table *m* mal:

```
k = 3; m = 20;
Table[Apply[Plus, RandomInteger[{1, 6}, k]], {m}]
Clear[k, m]
```

```
{10, 15, 7, 8, 13, 10, 15, 10, 15, 6, 10, 10, 13, 6, 8, 12, 11, 7, 8, 12}
```

1.3 Münzwurf

Fast ebenso bekannt wie das Würfeln ist der Münzwurf. Da sich viele Fragestellungen der Stochastik am Münzwurf einfach veranschaulichen lassen, wollen wir auch diesen zufälligen Vorgang näher untersuchen. Dabei kommen wir überein, den Ausgang "Zahl" stets mit 0 und den Ausgang "Adler" stets mit 1 zu charakterisieren. Außerdem setzen wir wieder voraus, dass wir es mit einer homogenen Münze, also einer Münze, bei der die beiden Realisierungen "Zahl" und "Adler" mit der gleichen Wahrscheinlichkeit auftreten, zu tun haben.

1.3.1 Beispiel: Unser Zufallsexperiment besteht im einmaligen Werfen einer Münze.

- Man simuliere dieses Zufallsexperiment
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Wir simulieren das einmalige Werfen einer Münze mit Hilfe von RandomInteger:

```
RandomInteger[]
```

```
1
```

1.3.2 Beispiel: Unser Zufallsexperiment besteht im einmaligen Werfen von *k* Münzen.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Wir simulieren das einmalige Werfen von *k* Münzen mit Hilfe von RandomInteger:

```
k = 5;
RandomInteger[{0, 1}, k]
Clear[k]
```

```
{0, 1, 0, 1, 0}
```

1.3.3 Beispiel: Unser Zufallsexperiment besteht im *m*-maligen Werfen von *k* Münzen.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Analog zu Beispiel 1.3.2 simulieren wir mit Hilfe von RandomInteger das einmalige Werfen von *k* Münzen. Diesen Vorgang wiederholen wir mit Hilfe von Table *m* mal:

```
k = 6; m = 10;
Table[RandomInteger[{0, 1}, k], {m}]
Clear[k, m]
```

```
{{1, 1, 1, 0, 1, 1}, {0, 0, 1, 0, 1, 0}, {1, 1, 0, 0, 0, 1},
 {1, 0, 0, 0, 0, 0}, {0, 1, 0, 0, 1, 1}, {0, 1, 1, 0, 1, 1},
 {1, 0, 1, 0, 0, 0}, {1, 1, 0, 1, 1, 1}, {1, 0, 0, 0, 1, 0}, {0, 1, 1, 1, 0, 0}}
```

1.3.4 Beispiel: Unser Zufallsexperiment besteht im *m*-maligen Werfen von *k* Münzen und der Feststellung, wie oft dabei jeweils das Ereignis "Adler" eingetreten ist.
- Man simuliere dieses Zufallsexperiment.
- Man beobachte wieder, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Analog zu Beispiel 1.3.2 simulieren wir zuerst das einmalige Werfen von *k* Münzen und ermitteln anschließend mit Hilfe von Apply und Plus, wie oft dabei das Ereignis "Adler" eingetreten ist. Diesen Vorgang wiederholen wir mit Hilfe von Table *m* mal:

```
k = 6; m = 20;
Table[Apply[Plus, RandomInteger[{0, 1}, k]], {m}]
Clear[k, m]
```

```
{4, 4, 2, 2, 4, 2, 3, 0, 4, 4, 3, 2, 3, 4, 2, 3, 2, 4, 4, 2}
```

1.4 Ziehen von Kugeln aus einer Urne

Urnenmodelle haben in der Stochastik deshalb eine herausragende Bedeutung, weil sich damit zahlreiche stochastische Vorgänge in übersichtlicher Weise und ohne belastendes Beiwerk beschreiben lassen. Bei diesen Urnenmodellen geht es dabei stets um das Ziehen von Kugeln aus Urnen oder das Verteilen von Kugeln auf Urnen. Wir befassen uns vorerst nur mit dem Ziehen von Kugeln aus einer Urne. Wird dabei die gezogene Kugel sofort wieder in die Urne zurückgelegt, so spricht man vom Ziehen mit Zurücklegen; wird die gezogene Kugel nicht mehr in die Urne zurückgelegt, so spricht man vom Ziehen ohne Zurücklegen. Wir nehmen dabei an, dass fair gezogen wird, also jede Kugel der Urne jeweils die gleiche Chance hat, gezogen zu werden.

Wir kommen überein, die m Kugeln der Urne mit den Zahlen 1, 2, …, m zu nummerieren. Die Liste {3, 1, 3, 2} entspricht damit der Realisierung "beim ersten Zug wird die Kugel mit der Nummer 3, beim zweiten Zug wird die Kugel mit der Nummer 1, beim dritten Zug wird die Kugel mit der Nummer 3 und beim vierten Zug wird die Kugel mit der Nummer 2 gezogen".

1.4.1 Beispiel: Unser Zufallsexperiment besteht im Ziehen von k Kugeln aus einer Urne mit m Kugeln, wobei die gezogene Kugel jeweils sofort wieder in die Urne zurückgelegt wird.
- Man simuliere das Ziehen mit Zurücklegen von k Kugeln aus einer Urne mit m Kugeln.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Beim Ziehen mit Zurücklegen ist wesentlich, dass wir es bei jedem neuen Zug mit dem zufälligen Ziehen einer Kugel aus einer Urne mit m Kugeln zu tun haben. Das Ziehen mit Zurücklegen von k Kugeln aus einer Urne mit m Kugeln lässt sich daher mit Hilfe von RandomInteger leicht simulieren:

```
k = 20; m = 25;
RandomInteger[{1, m}, k]
Clear[k, m]
```

```
{23, 7, 15, 11, 2, 19, 25, 17, 10, 1, 11, 1, 17, 24, 3, 10, 22, 2, 14, 3}
```

1.4.2 Beispiel: Unser Zufallsexperiment besteht im Ziehen von k Kugeln aus einer Urne mit $m \geq k$ Kugeln, wobei die gezogenen Kugeln nicht mehr in die Urne zurückgelegt werden.
- Man simuliere das Ziehen ohne Zurücklegen von k Kugeln aus einer Urne mit $m \geq k$ Kugeln.
- Man beobachte, wie sich die dabei erzeugte Realisierung von Versuch zu Versuch ändert.

▼

Lösung: Wir erzeugen zuerst mit Hilfe von Range die Liste {1, 2, …, m}. Mit Hilfe von RandomSample erzeugen wir anschließend eine pseudozufällige Permutation dieser Liste und wählen davon die ersten k Elemente aus:

```
k = 10; m = 25;
RandomSample[Range[m], k]
Clear[k, m]
```

```
{10, 22, 1, 12, 5, 3, 13, 24, 25, 20}
```

1.4.3 Beispiel: Das österreichische Lotto besteht im Ziehen ohne Zurücklegen von $k = 6$ Kugeln aus einer Urne mit $m = 45$ Kugeln, welche mit den Zahlen 1, 2, …, 45 nummeriert sind und dem anschließenden Anordnen der gezogenen Kugeln nach der Größe der sie tragenden Nummern.

- Man simuliere dieses Zufallsexperiment.
- Man beobachte, wie viele Versuche notwendig sind, bis erstmals die Kugel mit der Zahl 1 gezogen wird.

▼

Lösung: Analog zu Beispiel 1.4.2 simulieren wir das Ziehen ohne Zurücklegen von $k = 6$ Kugeln aus einer Urne mit $m = 45$ Kugeln und ordnen die dabei erzeugte Realisierung anschließend mit Hilfe von Sort der Größe nach:

```
k = 6; m = 45;
Sort[RandomSample[Range[m], k]]
Clear[k, m]
```

```
{10, 12, 23, 28, 39, 40}
```