Simulation von Zufallsexperimenten mit der TI-Nspire™-CX-CAS-Technologie

Wolfgang Häfner

In den Bildungsstandards und den Lehrplänen für das Fach Mathematik findet man Hinweise zur Simulation von Zufallsexperimenten.

Bildungsstandards für das Fach Mathematik Erster Schulabschluss (ESA) und Mittlerer Schulabschluss (MSA)
„Diese Leitidee *(Daten und Zufall)* umfasst zwei Säulen, die beschreibende Statistik und die Wahrscheinlichkeitsrechnung zur Modellierung von zufallsabhängigen Vorgängen und Risiken. Wahrscheinlichkeiten können als Prognosen von relativen Häufigkeiten bei zufallsabhängigen Vorgängen gedeutet werden, wodurch die beiden Säulen verknüpft werden."
„Die **stochastische Simulation** spielt bei der Verknüpfung eine wichtige Rolle. Der Umgang mit Daten und Zufallserscheinungen im Alltag und Zufallsexperimenten geschieht **auch unter Verwendung einschlägiger digitaler Mathematikwerkzeuge, hier vor allem Tabellenkalkulation** und Stochastiktools."
…..
„Die Schüler nutzen **Simulationen**, um stochastische Fragen zu entscheiden."

Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife
„In Ausweitung und Vertiefung stochastischer Vorstellungen der Sekundarstufe I umfasst diese Leitidee *(Daten und Zufall)* insbesondere den Umgang mit mehrstufigen Zufallsexperimenten, die Untersuchung und Nutzung von Verteilungen sowie einen Einblick in Methoden der beurteilenden Statistik, **auch mithilfe von Simulationen** und unter Verwendung einschlägiger Software."

„Die Schülerinnen und Schüler können Simulationen zur Untersuchung stochastischer Situationen verwenden."

Lehrplan für den Erwerb der allgemeinen Hochschulreife - Thüringen
Lehrplan 9/10
„Die Schülerinnen und Schüler können
Ideen und Ergebnisse zur Beschreibung, **Simulation** und Berechnung von Zufallsexperimenten formulieren, bewerten und präsentieren."
Lehrplan Qualifikationsphase der Oberstufe
„Die Schülerinnen und Schüler können Zufallsvorgänge simulieren und stochastische Zusammenhänge mit Hilfe von Simulationen begründen."

Kerncurriculum für das Gymnasium Schuljahrgänge 5-10 Mathematik Niedersachsen
Lehrplan 7/8
„Simulationen werden mit realen Objekten sowie **mithilfe digitaler Mathematikwerkzeuge** durchgeführt. Das Erleben der Variabilität fördert ein Verständnis für den Unterschied zwischen Wahrscheinlichkeit und relativer Häufigkeit sowie ein qualitatives Verständnis für das Gesetz der großen Zahlen."

Fachlehrplan Mathematik Gymnasium Bayern
Lehrplan 10
„Die Schülerinnen und Schüler simulieren Zufallsexperimente und bestimmen so Näherungswerte für Wahrscheinlichkeiten, die sie noch nicht berechnen können (z.B. zu den **„vertauschten Briefen"** oder zum „Ziegenproblem"), bzw. überprüfen berechnete Wahrscheinlichkeiten auf Plausibilität (z.B. zum **„Geburtstagsproblem"**)."

Dieser Beitrag hat zum Ziel, die Simulationen von ausgewählten Zufallsexperimenten mit Hilfe der TI-Nspire™-CX-CAS-Technologie zu erklären und zu systematisieren.
Deshalb werde ich mich auf Zufallsexperimente konzentrieren, bei denen verschiedenfarbige Kugeln in einer Urne gegeben sind und eine Ziehung von Kugeln mit bzw. ohne Zurücklegen erfolgt. Damit ist dann schon eine Reihe anderer stochastisch analoger Zufallsexperimente abgedeckt. Auch Bernoulli-Experimente und damit binomialverteilte Zufallsgrößen können auf diese Weise einbezogen werden. Da die oben angeführten Anforderungen alle Schülerinnen und Schüler, also auch die ohne spezielle Fähigkeiten im Programmieren, betreffen, habe ich den Schwerpunkt auf Simulationen ohne entsprechende Kenntnisse gelegt und die Programme in TI-Basic bzw. Python für das Problem „vertauschte Briefe" und das „ Geburtstagsproblem" erst am Ende dieses Beitrags in etwas kürzerer Form vorgestellt.

In diesem Zusammenhang möchte ich auch auf folgende Materialien zur Stochastik hinweisen:

- Dr. Hubert Langlotz, Dr. Wilfried Zappe
  „Beispiele zum Einsatz des TI-Nspire™ CAS in der Stochastik“
  Texas Instruments Education Technology 2011
- Wolfgang Häfner
  „Die Nutzung der TI-Nspire CAS App für iPad an Beispielen erklärt“
  Texas Instruments Education Technology 2023/24
  Teil 2: Kapitel 7: „Stochastik - Simulation von Zufallsexperimenten
  Tel 5: Kapitel 15: „Binomialverteilung“, Kapitel 16: „Augensummen“,
  Kapitel 19: „Approximation der Binomialverteilung durch die Normalverteilung“

Um Zufallsexperimente mit Hilfe der TI-Nspire-Technologie zu simulieren, ist die Verwendung einiger der TI-Nspire-Software immanenter Funktionen erforderlich.
Diese werden nachfolgend in Anlehnung an das Referenzhandbuch und hinsichtlich ihrer Nutzung in diesem Beitrag erläutert:

randsamp(*Liste,#anz[,noRepl]*) Die Funktion gibt eine Liste mit einer Zufallsstichprobe von *#anz* Versuchen aus Liste zurück. Die Ziehung erfolgt ohne *noRepl* bzw. bei *noRepl*=0 mit, bei *noRepl*=1 ohne Zurücklegen.
rand*(#anz*) Die Funktion gibt eine Liste zurück, die *#anz* Zufallswerte zwischen 0 und 1 enthält.
randInt(*UntereGrenze, ObereGrenze,#anz*) Die Funktion gibt eine Liste mit *#anz* ganzzahligen Zufallszahlen innerhalb der festgelegten Grenzen zurück.
randBin(*n,p,#anz*) Die Funktion gibt eine Liste mit *#anz* Zufallszahlen (Anzahl der Erfolge) aus einer durch n (Anzahl der Stufen des Bernoulli-Experiments) und p (Erfolgswahrscheinlichkeit) festgelegten Binomialverteilung zurück.
seq(*Ausdr,Var,Von,Bis*) Die Funktion erhöht *Var* von *Von* bis *Bis* um jeweils 1, wertet den Ausdruck *Ausdr* aus und gibt die Ergebnisse als Liste zurück.
sum(*Liste*) Die Funktion gibt die Summe der Elemente der Liste zurück.
product(*Liste*) Die Funktion gibt das Produkt der Elemente der Liste zurück.
ifFn(*BoolscherAusdruck, Wert_wenn_wahr, Wert_wenn_falsch*) Die Funktion wertet den Boolschen Ausdruck *BoolscherAusdruck* der Reihe nach für jedes Listenelement aus. Ergibt die Auswertung ‚wahr‘, wird *Wert_wenn_wahr* zurückgegeben. Falls die Auswertung ‚falsch‘ ergibt, wird *Wert_wenn_falsch* zurückgegeben.
mod(*Liste,Teiler*) Ist *Liste* eine Liste natürlicher Zahlen und *Teiler* ebenfalls eine natürliche Zahl größer Null, dann gibt diese Funktion die Liste der Reste bei ganzzahliger Division der Listenelemente durch *Teiler* zurück.
countIf(*Liste, Kriterien*) Die Funktion gibt die kumulierte Anzahl aller Elemente in der Liste zurück, die die festgelegten Kriterien erfüllen. countIf(liste,k) Anzahl der Listenelemente, die den Wert k haben countIf(liste,?>z) Anzahl der Listenelemente, die größer als z sind
augment(*Liste1, Liste2*) Die Funktion gibt eine neue Liste zurück, die durch Anfügen von Liste2 an das Ende von Liste1 erzeugt wurde.

Alle angeführten Funktionen finden Sie im Katalog.

Prinzipielle Vorüberlegungen
Ziel der angestrebten Simulationen ist die Ermittlung der relativen Häufigkeit eines Ereignisses bei einem mehrstufigen Zufallsexperiment bei einer möglichst großen Anzahl von Versuchswiederholungen.
Da das betrachtete Zufallsexperiment mehrstufig ist, kann man die Ergebnisse als Listen erfassen. Durch die Versuchswiederholungen erhält man also eine Liste von Listen.
Diese Liste ist für die Erfassung in einer Spalte der List-&-Spreadsheet-Applikation nicht geeignet, da in Zellen die Eingabe von Listen nicht vorgesehen ist.
Man kann aber die Elemente der Ergebnisliste eines Zufallsversuchs eineindeutig auf Zahlen abbilden, so dass man die Versuchswiederholungen als Liste von Zahlen auch in der List-&-Spreadsheet-Applikation erfassen kann. Mit Hilfe der ifFn-Funktion kann man dann unter Berücksichtigung der Aufgabenbedingungen untersuchen, ob das Ergebnis des Zufallsexperiments Element des betrachteten Ereignisses ist oder nicht. Falls das Ergebnis zum Ereignis gehört, wird ihm der Wert „1“, sonst der Wert „Null“ zugeordnet. Damit ist dann die Anzahl der Einsen die absolute Häufigkeit des betrachteten Ereignisses.
Es sei hier noch darauf hingewiesen, dass die maximale Länge einer Liste in der List-&-Spreadsheet-Applikation 2500 beträgt.
Da man also die eineindeutige Zuordnung der Ergebnisliste und einer Zahlenliste vor Verwendung der List-&-Spreadsheet-Applikation bewerkstelligen muss, sind Berechnungen erforderlich, die man am besten in einer Notes-Applikation durchführt, damit eine Änderung der Versuchsbedingungen leicht umgesetzt werden kann und um die Simulation beliebig oft wiederholen zu können, wobei man in der Notes-Applikation nicht auf 2500 Wiederholungen aber durch eine Ressourcenauslastung beschränkt ist.

Ziehen mit Zurücklegen
Beispiel 1:
In einer Urne befinden sich drei rote, zwei schwarze und fünf blaue Kugeln. Es werden zufällig nacheinander drei Kugeln mit Zurücklegen gezogen und dabei das Ereignis E betrachtet:
E: „Es werden zwei blaue und eine rote Kugel gezogen.“

Den Farben werden jeweils verschiedene Primzahlen zugeordnet, z.B. „Rot“ die 2, „Schwarz“ die 3 und „Blau“ die 5.
Über das Produkt der gezogenen Primzahlen erhält man dann eine eineindeutige Zuordnung. Man nutzt also die Eindeutigkeit der Primfaktorzerlegung.
Für unser Beispiel gilt:
$\{b,b,r\}\ oder\ \{b,r,b\}\ oder\ \{r,b,b\} \quad \leftrightarrow \quad 2\cdot 5^2 = 50$

Der Inhalt der Urne wird durch die Liste $liste \coloneqq \{2,2,2,3,3,5,5,5,5,5\}$ eindeutig abgebildet.
Mit $product(randSamp(liste,3))$ erhält man die dem Ergebnis der Ziehung zugeordnete Zahl.

Die Liste $erg \coloneqq seq(product\big(randSamp(liste,3)\big),i,1,anzw)$ enthält die Ergebniszahlen für *anzw* Wiederholungen des Zufallsexperiments.

Die Liste $treffer \coloneqq ifFn(erg = 50,1,0)$ enthält nur Nullen und Einsen.
Die Anzahl der Einsen ist die absolute Häufigkeit des Ereignisses E.
Mit $\frac{sum(treffer)}{1.\cdot anzw}$ erhält man die relative Häufigkeit.

Lösung mit ‚Notes‘:

liste:={2,2,2,3,3,5,5,5,5,5} ▸ {2,2,2,3,3,5,5,5,5,5}

anzw:=2500 ▸ 2500

erg:=seq(product(randSamp(liste,3)),i,1,anzw)

▸ {125,50,50,75,75,20,20,12,30,30,30,20,20,27,50,75,18,30,50,75,75,30,50,20,20,50,20,50,50,30,2(

treffer:=ifFn(erg=50,1,0)

▸ {0,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,1,0,1,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1

$\frac{sum(treffer)}{1.\cdot n}$ ▸ 0.2308

Natürlich könnte die Berechnung auch durch eine einzige Anweisung erfolgen:

$$\frac{sum(ifFn(seq(product(randSamp(liste,3)),i,1,anzw)=50,1,0))}{1.\cdot anzw}$$

Ich denke aber, dass die Aufteilung der Lösung in mehrere Lösungsschritte dem Verständnis förderlicher ist. Damit ist dann auch die Einbeziehung der Applikation ‚Lists & Spreadsheet‘ sinnvoll.

Lösung mit ‚Notes‘ und ‚Lists & Spreadsheet‘

	A erg	B treffer	C	D	E	F
=	=seq(product(randsamp(liste,3)),i,	=iffn(erg=50,1,0)				
1	50	1	0.224			
2	75	0				
3	18	0				
4	30	0				
5	20	0				

D

liste:={2,2,2,3,3,5,5,5,5,5} ▸ {2,2,2,3,3,5,5,5,5,5}

anzw:=2500 ▸ 2500

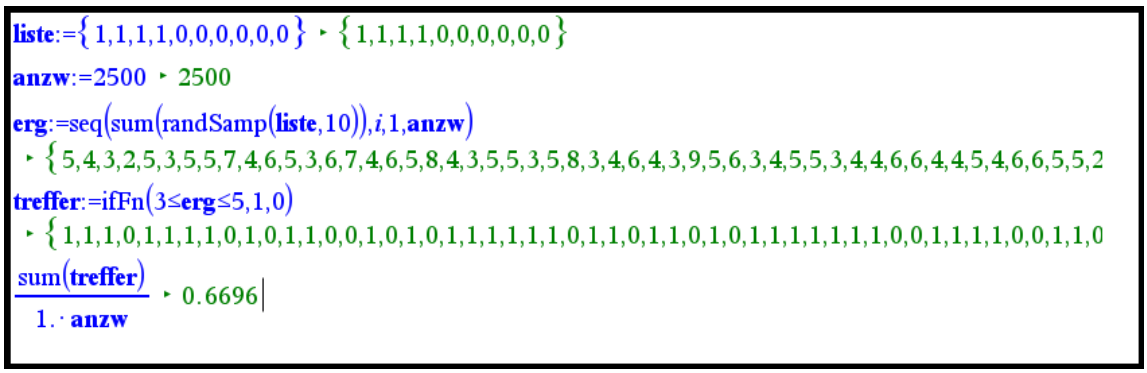
Theoretisch ermittelte Wahrscheinlichkeit:

$$3\cdot\frac{5}{10}\cdot\frac{5}{10}\cdot\frac{3}{10}=\frac{9}{40}=0{,}225$$

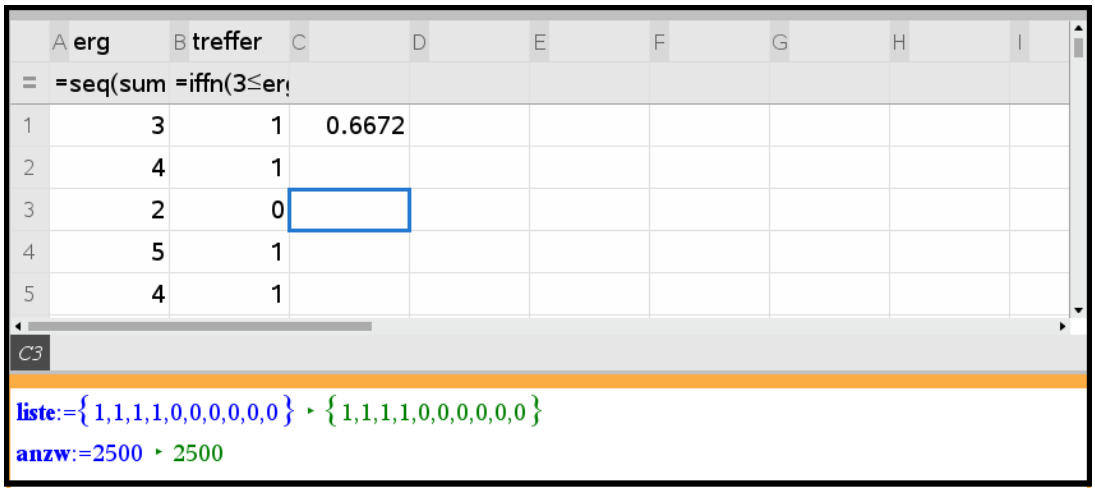
Beispiel 2:
In einer Urne befinden zehn Kugeln, von denen vier rot sind. Es werden zufällig nacheinander zehn Kugeln mit Zurücklegen gezogen und dabei das Ereignis E betrachtet:
E: „Es werden mindestens drei und höchstens fünf rote Kugeln gezogen.“

Ordnet man bei der Simulation dieses Zufallsversuchs den roten Kugeln die 1 und den anderen Kugeln die 0 zu, dann kann die Anzahl der Treffer über die Summe der zugeordneten Zahlen ermittelt werden.
Der Inhalt der Urne wird durch die Liste $liste := \{1,1,1,1,0,0,0,0,0,0\}$ eindeutig abgebildet.
Die Liste $erg := seq(sum(randSamp(liste, 10)), i, 1, anzw)$ enthält die jeweilige Anzahl der gezogenen roten Kugeln für *anzw* Wiederholungen.
Die Trefferliste ergibt sich hier mit $treffer := ifFn(3 \leq erg \leq 5,1,0)$.
Mit $\frac{sum(treffer)}{1. \cdot anzw}$ erhält man die relative Häufigkeit.

Lösung mit ‚Notes':



Lösung mit Notes und Lists & Spreadsheet:



Ziehen ohne Zurücklegen
Beispiel 3:
In einer Urne befinden sich drei rote, zwei schwarze und fünf blaue Kugeln. Es werden zufällig drei Kugeln ohne Zurücklegen gezogen und dabei das Ereignis E betrachtet:
E: „Es werden zwei blaue und eine rote Kugel gezogen."

Dieses Beispiel entspricht, bis auf die Tatsache, dass die gezogenen Kugeln nicht zurückgelegt werden, dem Beispiel 1.
Daher muss lediglich $randSamp(liste, 3)$ durch $randSamp(liste, 3,1)$ ersetzt werden.

Lösung mit ‚Notes‘:

liste:={2,2,2,3,3,5,5,5,5,5} ▸ {2,2,2,3,3,5,5,5,5,5}

anzw:=2500 ▸ 2500

erg:=seq(product(randSamp(liste,3,1)),i,1,anzw)

▸ {75,75,30,20,50,50,50,20,75,12,20,50,30,50,30,30,30,30,75,30,30,18,30,30,50,75,30,50,45,20,20,

treffer:=ifFn(erg=50,1,0)

▸ {0,0,0,0,1,1,1,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,1,0,1,0,0,0,0,1,0,1,1,0,0,0,0,1,0,0,0,0

$\frac{\text{sum(treffer)}}{1.\cdot \text{anzw}}$ ▸ 0.248

Lösung mit ‚Notes‘ und ‚Lists & Spreadsheet‘

	A erg	B treffer	C	D	E	F
=	=seq(product(randsamp(liste,3,1))	=iffn(erg=50,1,0)				
1	75	0	0.2512			
2	30	0				
3	50	1				
4	125	0				
5	45	0				

D

liste:={2,2,2,3,3,5,5,5,5,5} ▸ {2,2,2,3,3,5,5,5,5,5}

n:=2500 ▸ 2500

Theoretisch ermittelte Wahrscheinlichkeit:

$$3 \cdot \frac{5 \cdot 4 \cdot 3}{10 \cdot 9 \cdot 8} = \frac{1}{4} \qquad \frac{\binom{5}{2} \cdot \binom{3}{1}}{\binom{10}{3}} = \frac{1}{4}$$

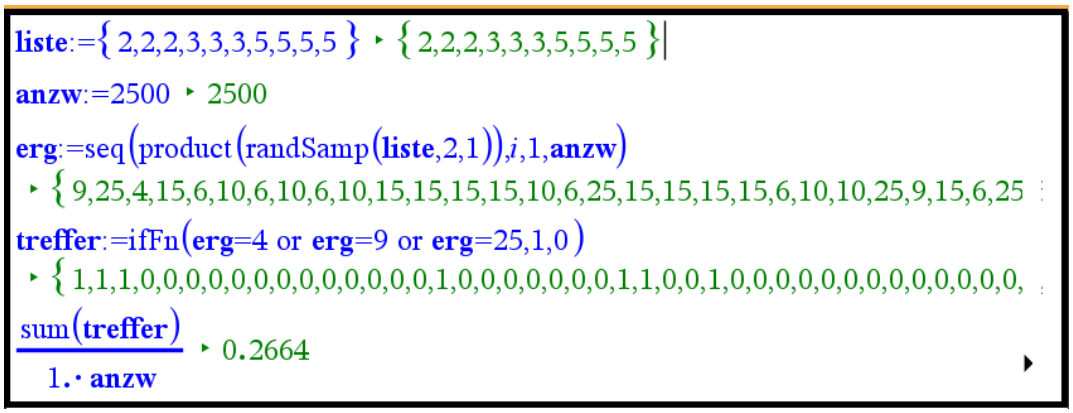
Beispiel 4:
In einer Urne befinden sich drei rote, drei schwarze und vier blaue Kugeln. Es werden zufällig zwei Kugeln ohne Zurücklegen gezogen und dabei das Ereignis E betrachtet:
E: „Die gezogenen Kugeln haben die gleiche Farbe.“

Die Überlegungen und das Vorgehen entspricht prinzipiell den bisherigen Beispielen.
Für Beispiel 4 gilt:
$\{r,r\} \leftrightarrow 2^2 = 4; \quad \{s,s\} \leftrightarrow 3^2 = 9; \quad \{b,b\} \leftrightarrow 5^2 = 25$

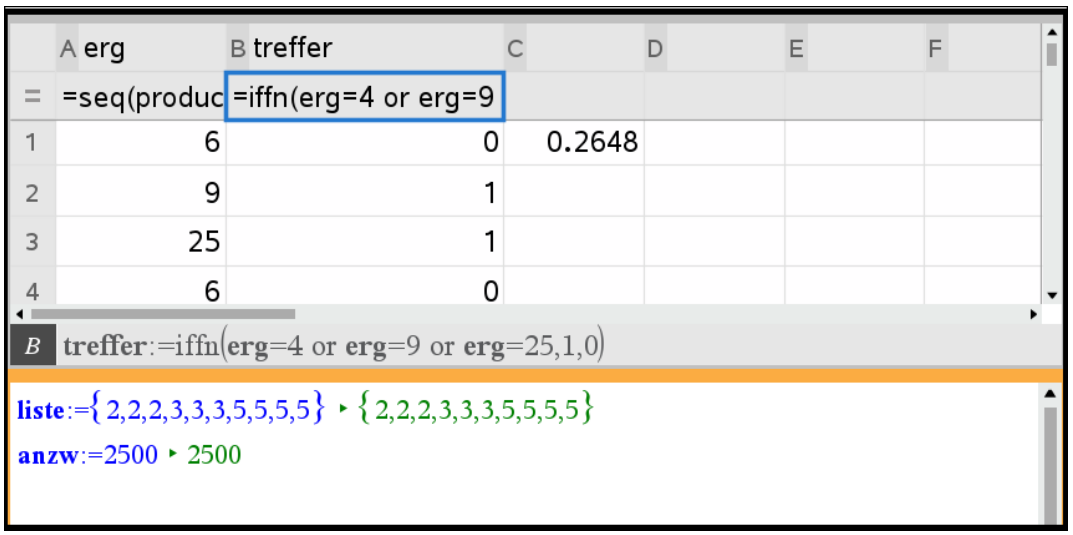
Der Inhalt der Urne wird durch die Liste $liste \coloneqq \{2,2,2,3,3,3,5,5,5,5\}$ eindeutig abgebildet.

Notwendige Änderungen gegenüber Beispiel 3:
Die Funktion $randSamp(liste, 3,1)$ muss durch $randSamp(liste, 2,1)$ ersetzt werden und $'erg = 50'$ durch $'erg = 4\ or\ erg = 9\ or\ erg = 25'$.

Lösung nur mit ‚Notes‘:



Lösung mit ‚Notes‘ und ‚Lists & Spreadsheet‘:



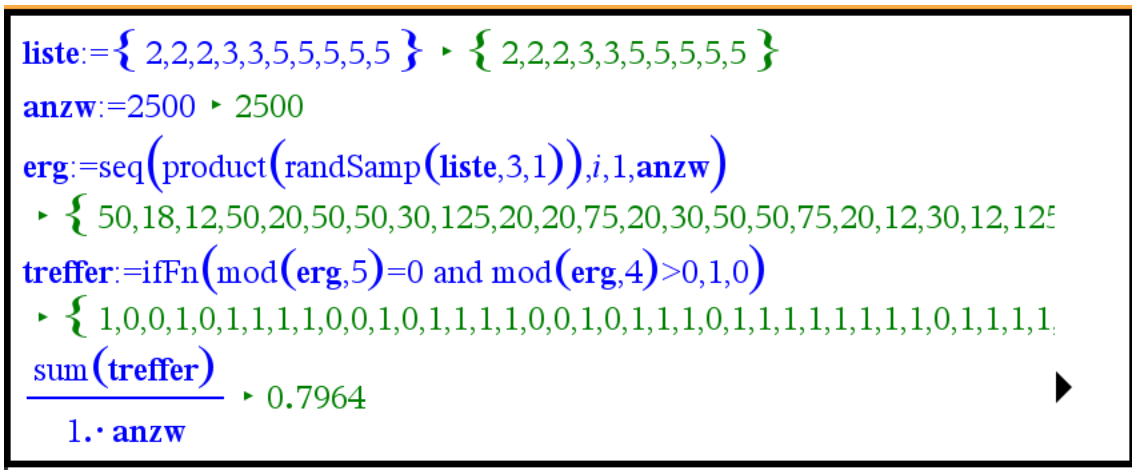
Theoretisch ermittelte Wahrscheinlichkeit:

$$\frac{3}{10}\cdot\frac{2}{9}+\frac{3}{10}\cdot\frac{2}{9}+\frac{4}{10}\cdot\frac{3}{9}=\frac{4}{15}=0{,}2\overline{6} \qquad 2\cdot\frac{\binom{3}{2}}{\binom{10}{2}}+\frac{\binom{4}{2}}{\binom{10}{2}}=\frac{4}{15}$$

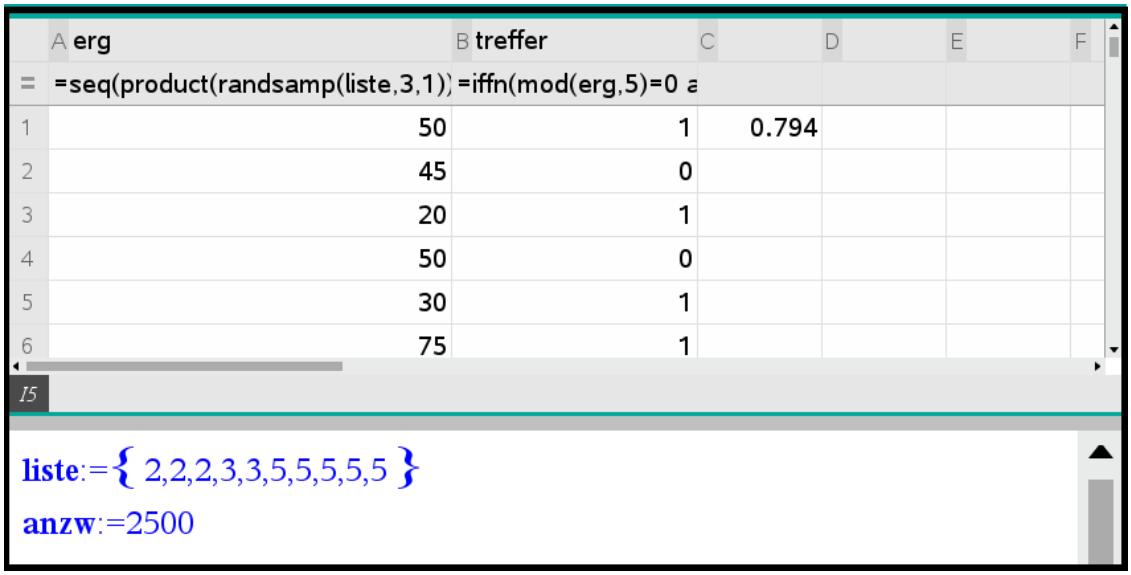
Beispiel 5:
In einer Urne befinden sich drei rote, zwei schwarze und fünf blaue Kugeln. Es werden zufällig drei Kugeln ohne Zurücklegen gezogen und dabei das Ereignis E betrachtet:
E: „Es wird mindestens eine blaue und höchstens eine rote Kugel gezogen.“

Für dieses Beispiel gilt:
Das dem Ereignis zugeordnete Produkt z muss durch 5, aber nicht durch 4 teilbar sein. Die Teilbarkeit durch 5 gewährleistet, dass unter den gezogenen Kugeln mindestens eine blaue ist. Bei einer Teilbarkeit durch 4 gäbe es mindestens zwei rote unter den gezogenen Kugeln. Die entsprechende Bedingung lautet also: $z\ mod\ 5 = 0\ \ and\ z\ mod\ 4 > 0$
Der Inhalt der Urne wird wie beim ersten Beispiel durch die Liste $liste := \{2,2,2,3,3,5,5,5,5,5\}$ eindeutig abgebildet.
Notwendige Änderungen gegenüber Beispiel 4:
Die Ergebnisse werden wieder mit $randSamp(liste, 3,1)$ ermittelt.
Für die Treffer gilt:
$treffer := ifFn(mod(erg, 5) = 0\ and\ mod(erg, 4) > 0,1,0)$

Lösung nur mit Notes:



Lösung mit ‚Notes' und ‚Lists & Spreadsheet':



Theoretisch ermittelte Wahrscheinlichkeit:

$$4 \cdot \frac{5 \cdot 4 \cdot 3}{10 \cdot 9 \cdot 8} + 3 \cdot \frac{5 \cdot 4 \cdot 2}{10 \cdot 9 \cdot 8} + 6 \cdot \frac{5 \cdot 3 \cdot 2}{10 \cdot 9 \cdot 8} + 3 \cdot \frac{5 \cdot 2 \cdot 1}{10 \cdot 9 \cdot 8} = \frac{19}{24} \approx 0{,}7917$$

$$\frac{\binom{5}{3}}{\binom{10}{3}} + \frac{\binom{5}{2} \cdot \binom{3}{1}}{\binom{10}{3}} + \frac{\binom{5}{2} \cdot \binom{2}{1}}{\binom{10}{3}} + \frac{\binom{5}{1} \cdot \binom{3}{1} \cdot \binom{2}{1}}{\binom{10}{3}} + \frac{\binom{5}{1} \cdot \binom{2}{2}}{\binom{10}{3}} = \frac{19}{24}$$

Simulation von n-stufigen Bernoulli-Experimenten mit der Erfolgswahrscheinlichkeit p

In Beispiel 2 liegt bereits ein zehnstufiges Bernoulli-Experiment mit der Erfolgswahrscheinlichkeit $p = 0{,}4$ vor.
Die Arbeit mit $randsamp(liste, n)$ ist also möglich, führt aber bei großem n, vielen Wiederholungen und manchen p-Werten zu unnötig großen Datenmengen. Daher stellen die weiter unten gezeigten Alternativen, insbesondere die beiden letzten, bessere Möglichkeiten der Simulation dar. Die Auswahl des Verfahrens sei aber jedem freigestellt.

Simulation eines $\boldsymbol{n}$-stufigen Bernoulli-Experiments mit $\boldsymbol{p = \frac{z}{d}}$ und $\boldsymbol{anzw}$ Wiederholungen, wobei die relative Häufigkeit für mindestens $\boldsymbol{a}$ und höchstens $\boldsymbol{b}$ Erfolge ermittelt werden soll.
Bespiel: $n = 24,\quad p = 0{,}29,\quad a = 4,\quad b = 8$

<u>Simulation mit Nutzung der Funktion *randsamp*</u>

Lösung mit ‚Notes':

```
z:=29 ▸ 29    d:=100 ▸ 100    p:=z/d ▸ 29/100    n:=24 ▸ 24    a:=4 ▸ 4    b:=8 ▸ 8
anzw:=2500 ▸ 2500
einsen:=seq(1,i,1,z) ▸ {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}
nullen:=seq(0,i,1,d−z)
 ▸ {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
liste:=augment(einsen,nullen)
 ▸ {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0▸
erg:=seq(sum(randSamp(liste,n)),i,1,anzw)
 ▸ {4,11,8,11,5,6,7,8,7,6,7,8,6,9,10,8,8,4,7,9,5,10,3,8,9,5,6,11,7,9,8,5,5,3,5,7,6,5,11,9,9,9,8,10,13,2,
treffer:=ifFn(a≤erg≤b,1,0)
 ▸ {1,0,1,0,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,0,1,0,0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,1,0,0,0,0,1,0,0,0,1,1,1,1,1,1
sum(treffer)/(1. · anzw) ▸ 0.7088
```

Erläuterung:
Die Liste $liste$ sollte bei Anwendung der Funktion $randsamp$ z Einsen und $(d - z)$ Nullen enthalten. Die Wahrscheinlichkeit, mit Zurücklegen eine Eins zu ziehen, ist dann nämlich $p = \frac{z}{d}$.
Die Liste $einsen \coloneqq seq(1, i, 1, z)$ ist eine Liste mit z Einsen.
Die Liste $nullen \coloneqq seq(0, i, 1, d - z)$ ist eine Liste mit $(d - z)$ Nullen
Die Funktion $augment(einsen, nullen)$ erzeugt eine Liste, bei der den Elementen der Liste $einsen$ die Elemente der Liste $nullen$ hinzugefügt werden.
Das weitere Vorgehen entspricht dem im Beispiel 2.
Die Wahrscheinlichkeit des betrachteten Ereignisses erhält man mit $binomCdf(n, p, a, b)$.

```
binomCdf(n,p,a,b) ▸ 0.70823
```

Simulation mit ‚Notes' und ‚Lists & Spreadsheet':

	A erg	B treffer	C	D	E	F	G	H	I
=	=seq(sum	=iffn('a≤er							
1	8	1	0.7048						
2	8	1							
3	6	1							
4	2	0							
5	7	1							

D

z:=29 ▸ 29 d:=100 ▸ 100 $p:=\frac{z}{d}$ ▸ $\frac{29}{100}$ n:=24 ▸ 24 a:=4 ▸ 4 b:=8 ▸ 8

anzw:=2500 ▸ 2500

einsen:=seq(1,i,1,z) ▸ {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}

nullen:=seq(0,i,1,d−z)

▸ {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

liste:=augment(einsen,nullen)

▸ {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

<u>Simulation mit Nutzung der Funktion $randInt(n)$:</u>

Lösung mit ‚Notes':

z:=29 ▸ 29 d:=100 ▸ 100 $p:=\frac{z}{d}$ ▸ $\frac{29}{100}$ n:=24 ▸ 24 a:=4 ▸ 4 b:=8 ▸ 8

anzw:=2500 ▸ 2500

erg:=seq(sum(ifFn(randInt(1,d,n)≤z,1,0)),i,1,anzw)

▸ {8,6,6,9,3,7,8,5,6,7,7,10,2,6,6,6,8,8,7,5,5,7,10,6,4,6,7,8,7,3,7,12,6,8,8,7,3,9,4,9,6,5,8,10,5,7,8,5,3,

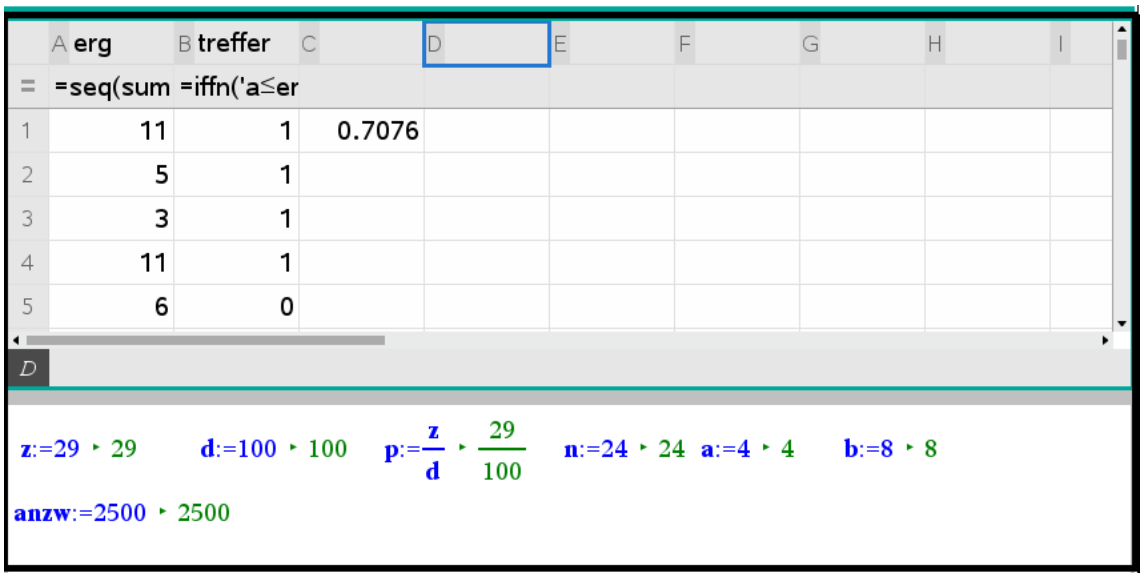
treffer:=ifFn(a≤erg≤b,1,0)

▸ {1,1,1,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,0,1,0,1,1,1,1,0,0,1,0,1,1,1,0,1,1,1,1,0,1,0,1

$\frac{\text{sum(treffer)}}{1. \cdot \text{anzw}}$ ▸ 0.7076

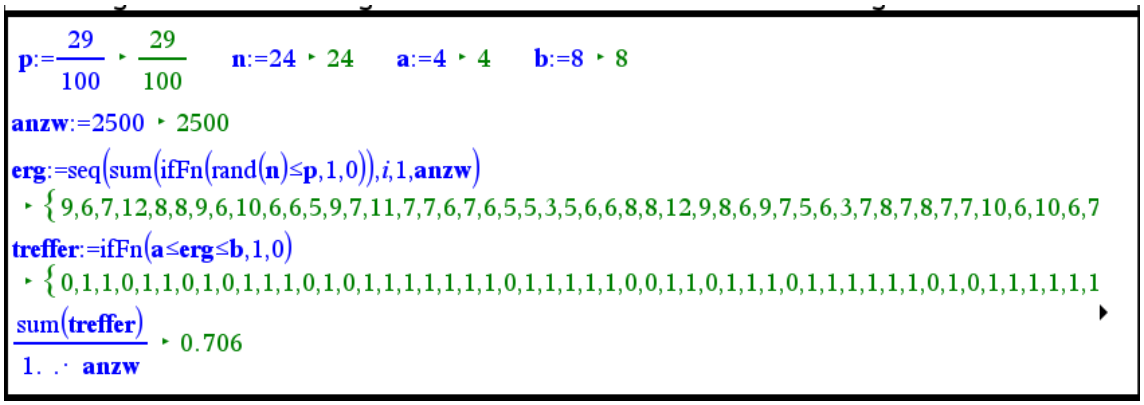
Erläuterung:
Die Funktion $randInt(1, d, n)$ gibt eine Liste mit n ganzen Zufallszahlen zwischen 1 und d zurück, die mit einer Wahrscheinlichkeit von $p = \frac{z}{d}$ kleiner gleich z sind. Die Eigenschaft „kleiner gleich z" kann damit als Erfolg mit der Erfolgswahrscheinlichkeit p betrachtet werden.
Die Liste $ifFn(randInt(1, d, n) \leq z, 1,0)$ enthält Einsen und Nullen, wobei die Anzahl der Einsen der Anzahl der Erfolge entspricht.
Die Liste $erg \coloneqq seq(sum(ifFn(randInt(1, d, n) \leq z, 1,0)), i, 1, anzw)$ ist die Liste der Anzahl der Erfolge bei $anzw$ Wiederholungen.

Lösung mit ‚Notes' und ‚Lists & Spreadsheet':



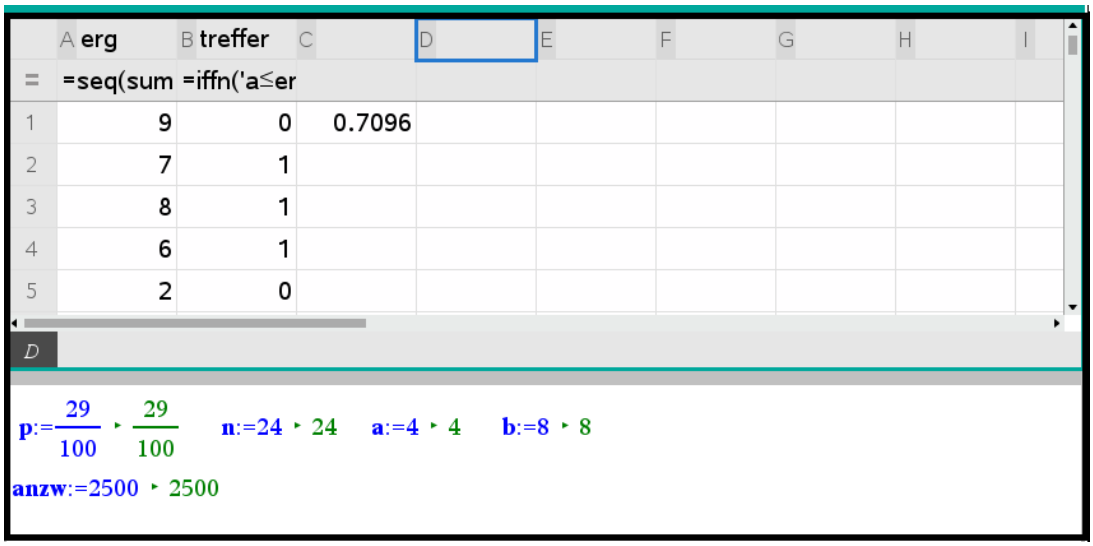
<u>Simulation mit Nutzung der Funktion $rand(n)$:</u>

Lösung mit ‚Notes':



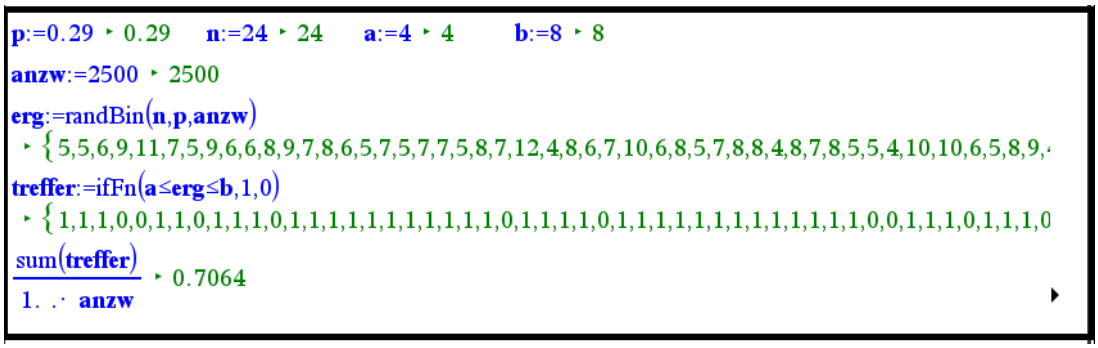
Erläuterung:
Die Funktion $rand(n)$ gibt eine Liste mit n Zufallszahlen zwischen 0 und 1 zurück, die mit einer Wahrscheinlichkeit von p kleiner oder gleich p sind. Die Eigenschaft „kleiner gleich p" kann damit als Erfolg mit der Erfolgswahrscheinlichkeit p betrachtet werden.
Die Liste $ifFn(rand(n) \leq p, 1,0)$ enthält Einsen und Nullen, wobei die Anzahl der Einsen der Anzahl der Erfolge entspricht.
Die Liste $erg := seq(sum(ifFn(rand(n) \leq p, 1,0)), i, 1, anzw)$ ist die Liste der Anzahl der Erfolge bei $anzw$ Wiederholungen.

Lösung mit ‚Notes' und ‚Lists & Spreadsheet':



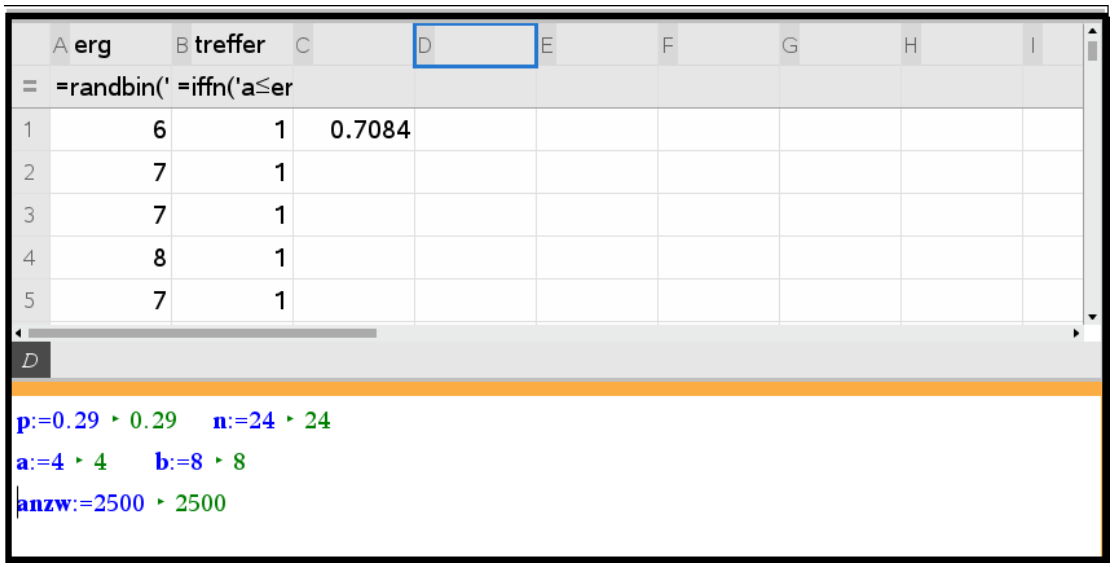
<u>Simulation mit Nutzung der Funktion $randBin$:</u>

Lösung mit ‚Notes':



Erläuterung:
Die Liste $erg := randBin(n, p, anzw)$ ist die Liste der Anzahl der Erfolge bei $anzw$ Wiederholungen eines n-stufigen Bernoulli-Experiments mit der Erfolgswahrscheinlichkeit p.

Lösung mit ‚Notes' und ‚Lists & Spreadsheet':



Das Problem der vertauschten Briefe

Mit Hilfe einer Simulation wird eine relative Häufigkeit für folgendes Ereignis ermittelt:
Ein zerstreuter Mitarbeiter verteilt in Eile n Briefe zufällig auf n bereits adressierte Briefumschläge. Danach befindet sich kein Brief im richtigen Umschlag.

Lösung mit TI-Nspire™-Basic:

Der Vorgang wird in einer Zählschleife mit $anzw$ Wiederholungen simuliert und die Liste $anzahl$ jeweils um die Zahl $treffer$ ergänzt. Die Variable $treffer$ hat den Wert 1, wenn das Ereignis eintritt, wenn nicht, hat sie den Wert 0.

Damit ist die Anzahl der Einsen in $anzahl$ die absolute und $\frac{sum(anzahl)}{anzw}$ die relative Häufigkeit des Ereignisses.
Die Liste $brief\_list$ ist die Liste der von 1 bis n durchnummerierten Briefe.
Die Liste $belegung$ eine zufällige Permutation von $brief\_list$ ohne Wiederholung.
Die Variable $vertauscht$ liefert die Anzahl der vertauschten Briefe.
Die Funktion $augment(anzahl,\{treffer\})$ ergänzt die Liste $anzahl$ durch die Liste $\{treffer\}$, die nur das Element $treffer$ enthält, indem sie $\{treffer\}$ an $anzahl$ anhängt

```
Define briefe1(n,anzw)=
Func
Local anzahl,brief_list,v,i,belegung,vertauscht,treffer
anzahl:={}
brief_list:=seq(i,i,1,n)
For v,1,anzw
  belegung:=randSamp(brief_list,n,1)
  vertauscht:=sum(seq(ifFn(belegung[i]=brief_list[i],0,1),i,1,n))
  treffer:=ifFn(vertauscht=5,1,0)
  anzahl:=augment(anzahl,{treffer})
EndFor
Return 1.·sum(anzahl)/anzw
EndFunc
```

briefe(5,4000)	0.3665
briefe(5,4000)	0.3705
briefe(5,4000)	0.3575
briefe(5,4000)	0.36475

"briefe" erfolg. gespeichert

```
Define briefe(n,anzw)=
Func
Local anzahl,brief_list,v,i,belegung,v
anzahl:={}
brief_list:=seq(i,i,1,n)
For v,1,anzw
  belegung:=randSamp(brief_list,n,1)
  vertauscht:=sum(seq(ifFn(belegung[,
  treffer:=ifFn(vertauscht=5,1,0)
  anzahl:=augment(anzahl,{treffer})
EndFor
Return 1.·sum(anzahl)/anzw
EndFunc
```

Lösung mit Python:

Das Programm funktioniert analog der in TI-Basic programmierten Funktion $briefe$.
Die Liste $belegung$ wird in einer Zählschleife mit n Wiederholungen erzeugt.
Dabei gibt $w = choise(brief\_list)$ ein Zufallselement w aus der Liste $brief\_list$ zurück, $brief\_list.remove(w)$ entfernt w aus $brief\_list$ und $belegung.append(w)$ hängt w an die Liste $belegung$ an.
Die Variablen $brief\_list$, $vertauscht$ und $treffer$ werden auf einer Notes-Seite erzeugt und dann im Programm verwendet.
Die Funktion $anzahl.append(treffer)$ ergänzt die Liste $anzahl$ um den jeweiligen Wert von $treffer$.

briefe.py erfolgreich gespeichert

```
from ti_system import *
from random import *
anzw=int(input("Anzahl der Versuchswiederholungen: "))
n=int(input("Anzahl der Briefe: "))
def briefe():
    store_value("n",n)
    anzahl=[]
    for v in range(anzw):
        belegung=[]
        brief_list=recall_list("brief_list")
        for i in range(n):
            w=choice(brief_list)
            brief_list.remove(w)
            belegung.append(w)
        store_list("belegung",belegung)
        treffer=recall_value("treffer")
        anzahl.append(treffer)
    w=sum(anzahl)/anzw
    print("relative Häufigkeit: ",w)
briefe()
```

Python-Shell

```
>>>#Running briefe.py
>>>from briefe import *
Anzahl der Versuchswiederholungen: 4000
Anzahl der Briefe: 5
relative Häufigkeit:  0.3535
>>>
>>>#Running briefe.py
>>>from briefe import *
Anzahl der Versuchswiederholungen: 4000
Anzahl der Briefe: 5
relative Häufigkeit:  0.369
>>>#Running briefe.py
>>>from briefe import *
Anzahl der Versuchswiederholungen: 4000
Anzahl der Briefe: 5
relative Häufigkeit:  0.3625
>>>
```

```
brief_list:=seq(i,i,1,n) ▸ {1,2,3,4,5}
belegung ▸ {2,1,5,4,3}
vertauscht:=sum(seq(ifFn(belegung[i]=brief_list[i],0,1),i,1,n)) ▸ 4
treffer:=ifFn(vertauscht=5,1,0) ▸ 0
```

Die Simulation erfolgte im dargestellten Beispiel mit 5 Briefen und 4000 Wiederholungen des Zufallsversuchs:

Die Wahrscheinlichkeit für das Ereignis erhält man dafür mit

$$\sum_{k=0}^{5}\left(\frac{(-1)^k}{k!}\right)=\frac{11}{30}=0{,}3\overline{6}$$

Das Geburtstagsproblem

Mit Hilfe einer Simulation wird eine relative Häufigkeit für folgendes Ereignis ermittelt:
In einer Gruppe von n Personen haben mindestens k Personen am selben Tag Geburtstag.

Lösung mit TI-Nspire™-Basic:

Der Vorgang wird in einer Zählschleife mit $anzw$ Wiederholungen simuliert und die Liste $treffer$ jeweils um die Zahl $erfolg$ ergänzt. Die Variable $erfolg$ hat den Wert 1, wenn das Ereignis eintritt, wenn nicht, hat sie den Wert 0.

Damit ist die Anzahl der Einsen in $treffer$ die absolute und $\frac{sum(treffer)}{anzw}$ die relative Häufigkeit des Ereignisses.
Die Liste geb_p enthält die n Geburtstage aller Personen der Gruppe.
Die Liste $anzahl$ gibt für jeden Tag des Jahres in der Reihenfolge der Tage die jeweilige Anzahl der Personen an, die an diesem Tag Geburtstag haben.
(Es werden 365 Tage berücksichtigt.)

```
"geburtstag" erfolg. gespeichert
Define geburtstag(k,anzw,n)=
Func
Local i,v,geb_p,anzahl,erfolg,treffer,w
treffer:={}
For v,1,anzw
  geb_p:=randInt(1,365,n)
  anzahl:=seq(countIf(geb_p,i),i,1,365)
  erfolg:=when(countIf(anzahl,?>k-1)≥1,1,0)
  treffer:=augment(treffer,{erfolg})
EndFor
Return 1.·sum(treffer)/anzw
EndFunc
```

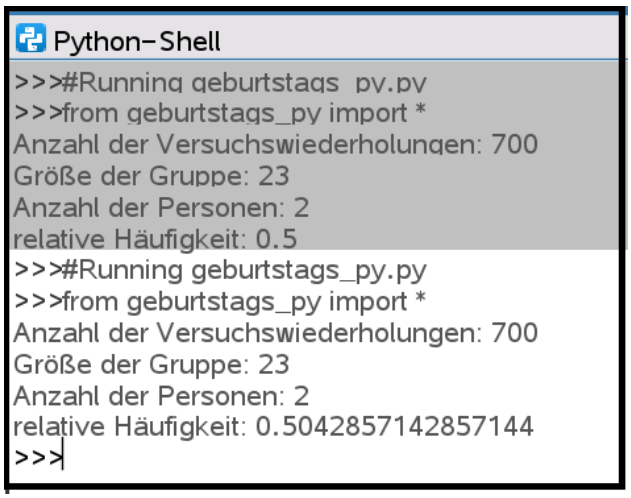
$geburtstag(2,700,23)$	0.505714
$geburtstag(2,700,23)$	0.515714
$geburtstag(2,700,23)$	0.505714
$geburtstag(2,700,23)$	0.485714
$geburtstag(2,700,23)$	0.501429

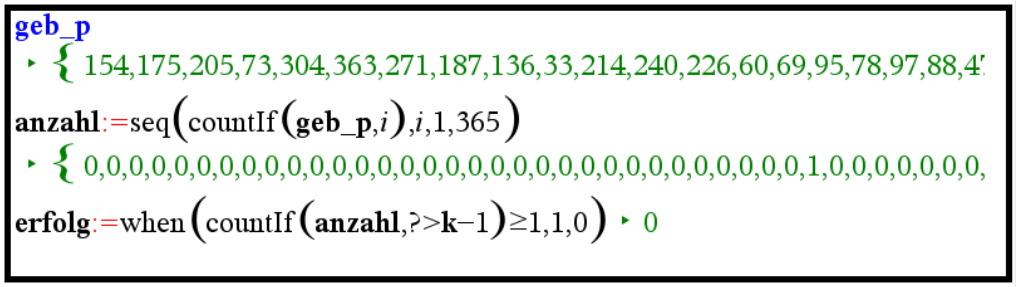
Lösung in Python:

Das Programm funktioniert analog der in TI-Basic programmierten Funktion $geburtstag$.
Die Liste geb_p wird in einer Zählschleife mit n Wiederholungen erzeugt. Die Liste wird jeweils durch eine ganze Zahl zwischen 1 und 365 (Geburtstag) ergänzt.
Die Variablen $anzahl$ und $erfolg$ werden auf einer Notes-Seite definiert, die Variable $erfolg$ dann im Programm verwendet.

geburtstags_py.py

```
from ti_system import *
from random import *
anzw=int(input("Anzahl der Versuchswiederholungen: "))
n=int(input("Größe der Gruppe: "))
k=int(input("Anzahl der Personen: "))
def geburtstag():
    store_value("k",k)
    treffer=[]
    for v in range(anzw):
        geb_p=[]
        for i in range(n):
            geb_p.append(randint(1,365))
        store_list("geb_p",geb_p)
        erfolg=recall_value("erfolg")
        treffer.append(erfolg)
    w=sum(treffer)/anzw
    print("relative Häufigkeit:",w)
geburtstag()
```





Die Simulation erfolgte im dargestellten Beispiel mit einer Gruppe von 23 Personen und es sollen mindestens 2 Personen am selben Tag Geburtstag haben.
Der Zufallsversuch wurde 700-mal wiederholt.

Die Wahrscheinlichkeit, dass von 23 Personen mindestens zwei am selben Tag Geburtstag haben, beträgt

$$1-\frac{\frac{365!}{(365-23)!}}{365^{23}}=1-\frac{nPr(365,23)}{365^{23}}\approx 0{,}507$$

Quellen:

1. Bildungsstandards für das Fach Mathematik Erster Schulabschluss (ESA) und Mittlerer Schulabschluss (MSA),
   Beschluss der Kultusministerkonferenz vom 15.10.2004 und vom 04.12.2003, i.d.F. vom 23.06.2022

2. Bildungsstandards im Fach Mathematik für die Allgemeine Hochschulreife
   Beschluss der Kultusministerkonferenz vom 18.10.2012

3. Kerncurriculum für das Gymnasium Schuljahrgänge 5-10 Mathematik Niedersachsen
   Herausgegeben vom Niedersächsischen Kultusministerium (2015)

4. Lehrplan für den Erwerb der allgemeinen Hochschulreife Mathematik,
   Thüringer Ministerium für Bildung, Jugend und Sport, 2018

5. Fachlehrplan Mathematik Gymnasium 10 Bayern
   https://www.lehrplanplus.bayern.de/fachlehrplan/gymnasium/10/mathematik

6. Dr. Hubert Langlotz, Dr. Wilfried Zappe
   „Beispiele zum Einsatz des TI-Nspire™ CAS in der Stochastik“
   Texas Instruments Education Technology 2011

7. Geburtstagsparadoxon
https://de.wikipedia.org/wiki/Geburtstagsparadoxon

8. Fixpunktfreie Permutation
https://de.wikipedia.org/wiki/Fixpunktfreie_Permutation

9. TI-Nspire™ CX CAS Referenzhandbuch,
Texas Instruments 2023

10. TI-Nspire™ Python Programmierhandbuch,
Texas Instruments 2021