

Simulaties en tellen met de TI-84

De TI-84 kan je behulpzaam zijn bij het bepalen van kansen. Hij kan simulaties van kansexperimenten uitvoeren en je helpen bij het tellen van mogelijkheden.

Loop eerst het practicum: 'Basistechnieken TI-84' door.

Inhoud

- [Simulaties](#)
 - [Werpen met dobbelstenen simuleren](#)
 - [Permutaties en combinaties](#)
-

Simulaties

Het werpen met een dobbelsteen kun je simuleren met **toevalsgetallen**. Bij de TI-84 vind je de 'randomizer' (toevalsgetallenmaker) door

- **MATH** te toetsen en dan naar de tab [PROB] te gaan met de pijltjestoetsen;
- vervolgens [rand] te kiezen;
- en dan op **ENTER** te blijven drukken.

Je krijgt zo toevalsgetallen tussen 0 en 1 (in tien decimalen).

Als je toevalsgetallen tussen 0 en 2 wilt, dan vermenigvuldig je ze met 2. In het rekenschermb doe je: **2****×****MATH****[PROB]****[rand]****ENTER**.

Meestal heb je echter **gehele toevalsgetallen** nodig (bijvoorbeeld bij de dobbelsteen de getallen 1 t/m 6). Die kun je krijgen door 'integer' (geheel getal) te gebruiken. De integer-routine laat gewoon alle decimalen weg, dus van 0,78456... maakt deze routine gewoon 0: $\text{int}(0,78456) = 0$. Maar $\text{int}(2 \cdot 0,78456) = \text{int}(1,56912) = 1$.

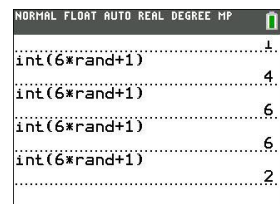
Bij de TI-84 vind je 'integer' zo:

- **MATH**{NUM}**[int]**

Als je $\text{int}(\mathbf{2} \times [\text{rand}])$ in het scherm zet en je geeft steeds **ENTER**, dan krijg je de toevalsgetallen 0 en 1.

Probeer maar...

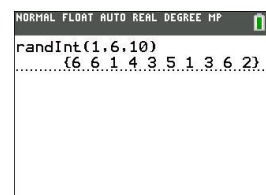
Het werpen met de dobbelsteen kun je nu simuleren door $\text{int}(\mathbf{6} \times [\text{rand}] + 1)$ in je rekenschermb te zetten en dan op **ENTER** te blijven drukken. In het plaatje hiernaast zie je dat gebeuren.



<code>int(6*rand+1)</code>	1
<code>int(6*rand+1)</code>	4
<code>int(6*rand+1)</code>	6
<code>int(6*rand+1)</code>	6
<code>int(6*rand+1)</code>	2

De TI-84 kent echter een routine om dit veel sneller te doen: `randInt()`. Voor de simulatie van 10 keer werpen met een dobbelsteen ga je dan zo te werk:

- **MATH****[PROB]****[randInt]**
- Een nieuw venster opent zich nu. Voer bij *lower* de minimale waarde van de dobbelsteen in, 1 dus. Voer bij *upper* de maximale waarde in, 6 dus. Bij *n* voer je het aantal simulaties dat je wilt uitvoeren in, 10 dus. Ga vervolgens naar *paste* en druk op **ENTER**.



<code>randInt(1,6,10)</code>
{6, 6, 1, 4, 3, 5, 1, 3, 6, 2}

De uitdrukking `randInt(1,6,10)` verschijnt nu in je rekenschermb. Voer deze uit. Dat levert je gehele toevalsgetallen vanaf 1 t/m 6 en wel ineens 10 stuks.



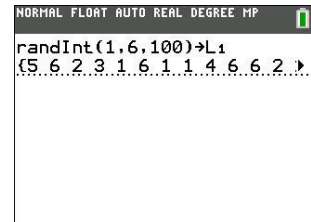
Werpen met dobbelstenen simuleren

Om met behulp van simulaties kansen te bepalen, moet je gemakkelijk kunnen **tellen hoe vaak elk getal in je simulatie voor komt**. Je zet dan je toevalsgetallen in een lijst.

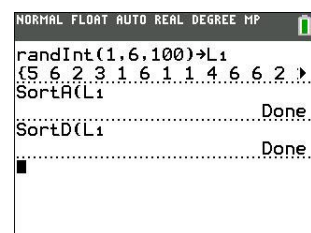
Werpen met één dobbelsteen

Stel je voor dat je 100 keer met een dobbelsteen gooien wilt simuleren en zo de kans wilt bepalen op het gooien van een 5. Je doet dan het volgende:

- Voer dezelfde simulatie uit als hiervoor, maar voer voor de n dit keer 100 in plaats van 10.
- Vertel de rekenmachine vervolgens dat dit in lijst L_1 moet door er achter te tikken:
`STO►[L1](2nd[1])ENTER`
- Alle toevalsgetallen staan nu in L_1 (lijst 1), je kunt dat zien door te tikken:
`STAT>EditENTER`
- Vervolgens kun je de lijst sorteren:
 - van klein naar groot:
`STAT[SortA()][ENTER][L1](2nd[1])ENTER`
(even wachten op de melding: Done)
SortA komt van Sort Ascending, sorteer oplopend.
 - van groot naar klein met SortD(
SortD komt van Sort Descending, sorteer aflopend.
- In de gesorteerde lijst kun je gemakkelijk tellen hoe vaak de vijf voorkomt van de 100 'worpen'.

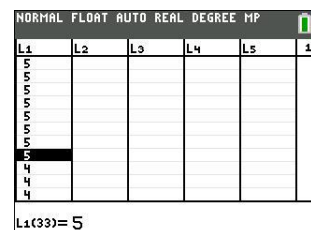


```
NORMAL FLOAT AUTO REAL DEGREE MP
randInt(1,6,100)→L1
{5 6 2 3 1 6 1 1 4 6 6 2 ...}
```



```
NORMAL FLOAT AUTO REAL DEGREE MP
randInt(1,6,100)→L1
{5 6 2 3 1 6 1 1 4 6 6 2 ...}
SortA(L1
SortD(L1
Done
Done
```

In de lijst hiernaast is de eerste 5 nummer 20 van de lijst en de laatste 5 was nummer 33. Er kwam dus 14 keer een vijf voor. De kans op 5 was daarom in deze simulatie $0,14$.



	L1	L2	L3	L4	L5	1
1	5					
2	6					
3	2					
4	3					
5	1					
6	6					
7	1					
8	1					
9	4					
10	6					
11	6					
12	2					
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33	5					
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						

L1(33)= 5

Nog eenvoudiger is het om eerst een **staafdiagram** te maken bij lijst L_1 via StatPlot. Hoe je een staafdiagram bij een lijst maakt vind je in het practicum 'Statistiek en de TI-84'.

Voer zelf zo'n simulatie uit.

Werpen met twee dobbelstenen

Als je bij het werpen met twee dobbelstenen de kans wilt bepalen op een bepaald aantal ogen dat op beide stenen samen boven komt te liggen, hebben niet alle mogelijkheden een gelijke waarschijnlijkheid. Bij je simulatie moet je daarmee rekening houden: je simuleert elke dobbelsteen afzonderlijk en telt dan de uitkomsten bij elkaar. Een simulatie van 100 worpen met twee dobbelstenen gaat zo:

- Eerst voer je `randInt(1,6,100)` uit. Zet het resultaat in L_1 .
- Vervolgens voer je `randInt(1,6,100)` opnieuw uit. Zet het resultaat in L_2 .
- Vervolgens zet je in het rekenscherm: $L_1 + L_2$ en `ENTER`.
- Het resultaat stop je vervolgens in L_3 .
- De simulatie van het werpen met twee dobbelstenen staat nu in L_3 . Deze lijst kun je sorteren en je kunt er een staafdiagram van maken. Met `TRACE` kun je nu gemakkelijk alle frequenties aflezen.

Voer zelf zo'n simulatie uit.

Dit is natuurlijk gemakkelijk uit te breiden tot het werpen met drie dobbelstenen, of vier munten, etc. Zolang het maar niet over al te grote aantallen gaat...



Permutaties en combinaties

Het aantal **permutaties** van 6 elementen is het totale aantal mogelijke verwisselingen als alle 6 elementen verschillend van elkaar zijn.

Dat aantal permutaties is: $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 6!$.

De TI-84 kan $6!$ op de volgende manier berekenen zonder de hele vermenigvuldiging in te tikken:

- Voer eerst de 6 in.
- Toets daarna: `[MATH][PROB][4](!)[ENTER]`.

Je ziet: $6! = 720$.

Bij het aantal **permutaties** van bijvoorbeeld 4 uit 10 gaat het om de mogelijke keuzes van 4 onderling verschillende elementen uit 10 verschillende elementen, dus om $10 \times 9 \times 8 \times 7 = 10! / 6!$.

Je kunt dat met deze rekenmachine als volgt berekenen:

- `[1][0][MATH][PROB][nPr][4][ENTER]`

Je vindt: 5040. Ga na dat dit hetzelfde is als $10 \times 9 \times 8 \times 7$.

Het aantal **combinaties** van 4 uit 10 is $10! / 4! / 6!$. Je schrijft het als $\binom{10}{4}$.

Dit is bijvoorbeeld het aantal uitkomsten als je uit 10 verschillende elementen er zonder terugleggen vier kiest, en daarbij niet op de uiteindelijke volgorde let.

Het is ook het aantal uitkomsten als je uit een groep van 10 elementen die bestaat uit 4 gelijke elementen en 6 gelijke andere elementen er zonder terugleggen vier kiest en daarbij wel op volgorde let.

Met de TI-84 kun je het aantal combinaties van 4 uit 10 zo berekenen:

- `[1][0][MATH][PROB][nCr][4][ENTER]`

Je vindt: 210. Ga na dat dit hetzelfde is als $10! / 4! / 6!$ en ook als $10! / (4! \times 6!)$.

Wanneer je het aantal mogelijke uitkomsten moet berekenen als je zonder terugleggen kiest uit 10 elementen waarbij een groep van 2 onderling gelijke, van 3 onderling gelijke andere en van 5 onderling gelijke nog andere elementen ontstaat, dan bereken je:

- $\binom{10}{2} \cdot \binom{8}{3}$ of
- $10! / 2! / 3! / 5!$ of $10! / (2! \times 3! \times 5!)$

