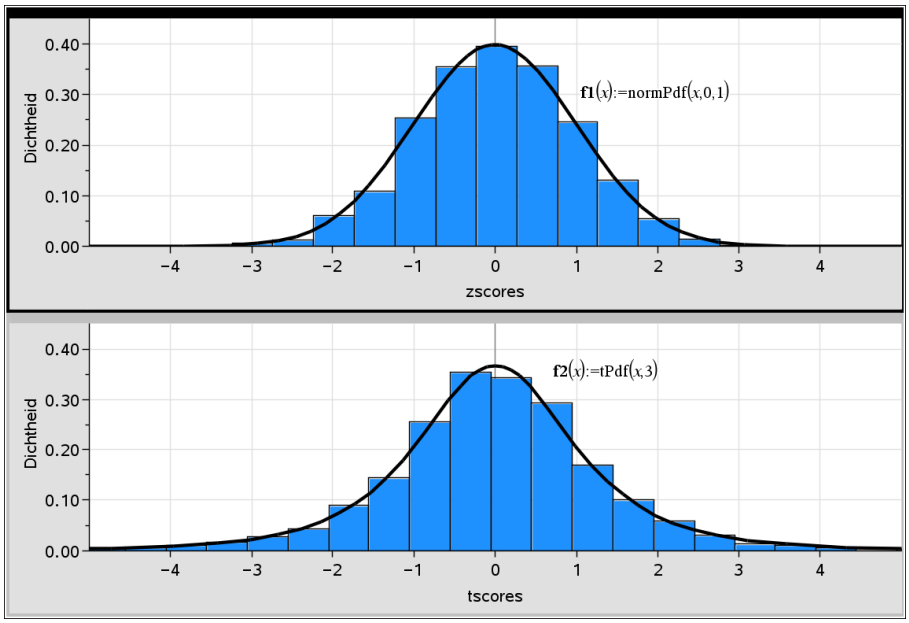
T³ VLAANDEREN

18de T³ Vlaanderen Symposium Oostende

24 & 25 augustus 2015

Kansverdelingen en simulaties met TI-Nspire CAS

Guido Herweyers



TEXAS INSTRUMENTS

KATHOLIEKE UNIVERSITEIT LEUVEN

Kansverdelingen en simulaties met TI-Nspire CAS

Guido Herweyers
VIVES en KU Leuven Technologiecampus Oostende

Inleiding

Wat is de exacte kansverdeling van de som van het aantal ogen als je werpt met 2, 3, 4, 5,… dobbelstenen en hoe kun je die kansverdeling benaderen door simulatie?
Hoe kun je de centrale limietstelling illustreren? Hoe kun je nieuwe kansverdelingen ontdekken via simulatie en wat is de meerwaarde van simulaties voor het begrijpen van toevalsprocessen? Deze onderwerpen komen hier aan bod.
Voor details over het gebruik van TI-Nspire verwijzen we naar cahier 28, www.t3vlaanderen.be

1 simuleren met TI-Nspire

1.1 Lukrake getallen genereren

Open een notitiepagina. Onder het menu *kansen, willekeurig*, vindt men de commando's om lukrake getallen te genereren.
Typ eerst het commando *randseed* in een math box gevolgd door een spatie en een natuurlijk getal, om een nieuwe reeks van lukrake getallen te beginnen.

```
Lukrake getallen genereren
RandSeed 125 ▸ Gereed

Een lukraak getal in het interval [0,1] : rand() ▸ 0.195097
Een lijst van 5 lukrake getallen in het interval [0,1]:
rand(5) ▸ {0.987567,0.169998,0.65364,0.836274,0.615509}
Een lukraak getal in het interval [3,7]: 3+4·rand() ▸ 6.55327

Werp een dobbelsteen: randInt(1,6) ▸ 5
Werp 10 dobbelstenen: randInt(1,6,10) ▸ {6,1,1,4,5,1,3,3,2,3}

Aantal keer kop bij 10 keer werpen van een muntstuk: randBin(10,0.5) ▸ 4
Herhaal dit 5 keer: randBin(10,0.5,5) ▸ {4,5,3,5,5}

Een lukraak getal uit een normale verdeling met gemiddelde 175 en standaardafwijking 10:
randNorm(175,10) ▸ 182.886
Een steekproef van 5 dergelijke getallen:
randNorm(175,10,5) ▸ {191.346,179.306,168.227,157.318,176.851}
```

Een steekproef trekken uit een populatie van numerieke of categorische data met het commando randsamp:

populatie:={10,20,30,30,30,50} ▸ {10,20,30,30,30,50}

Een steekproef met terugleggen:

randSamp(**populatie**,10) ▸ {20,30,50,30,10,30,20,30,30,20}

10 keer een roulette−rad draaien

roulette:=seq(*k*,*k*,0,36)
▸ {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36}

randSamp(**roulette**,10) ▸ {32,13,27,2,34,22,33,29,4,35}

Een steekproef zonder terugleggen:

randSamp(**populatie**,3,1) ▸ {30,10,30}

Een lottotrekking

lotto:=seq(*k*,*k*,1,45)
▸ {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45}

randSamp(**lotto**,6,1) ▸ {1,34,14,24,40,33}

Een simulatie (een uitdrukking waarin *rand…* optreedt) wordt automatisch opnieuw uitgevoerd door op de definitie van de uitdrukking te klikken en op *enter* te duwen (of gelijktijdig *Ctrl enter* voor snelle opeenvolgende herhalingen).

1.2 Data genereren uit een continue kansverdeling

Hoe kun je data genereren uit continue kansverdelingen die niet uniform of normaal verdeeld zijn?

Beschouw een continue kansverdeling met dichtheidsfunctie $f(x)$. We wensen hieruit een steekproef van grootte *n* te simuleren.

Bereken eerst de cumulatieve verdelingsfunctie $y = F(x) = P(X \le x)$ en de inverse functie $x = F^{-1}(y)$, genereer vervolgens met rand(*n*) een steekproef $\{y_1, y_2, y_3, \ldots, y_n\}$ uit de uniforme verdeling op het interval [0,1] en bereken $\{F^{-1}(y_1), F^{-1}(y_2), F^{-1}(y_3), \ldots, F^{-1}(y_n)\}$, dit is de gevraagde steekproef !

Inderdaad, als *Y* een toevalsvariabele is met een uniforme verdeling op [0,1] en $X = F^{-1}(Y)$, dan geldt $P(X \le x) = P\left(F^{-1}(Y) \le x\right) = P\left(Y \le F(x)\right) = F(x)$.

Data genereren uit een continue kansverdeling

Genereer data uit de verdeling met **dichtheidsfunctie** $f(x):=\begin{cases}-2\cdot x+2, & 0\le x\le 1\\ 0, & \text{else}\end{cases}$

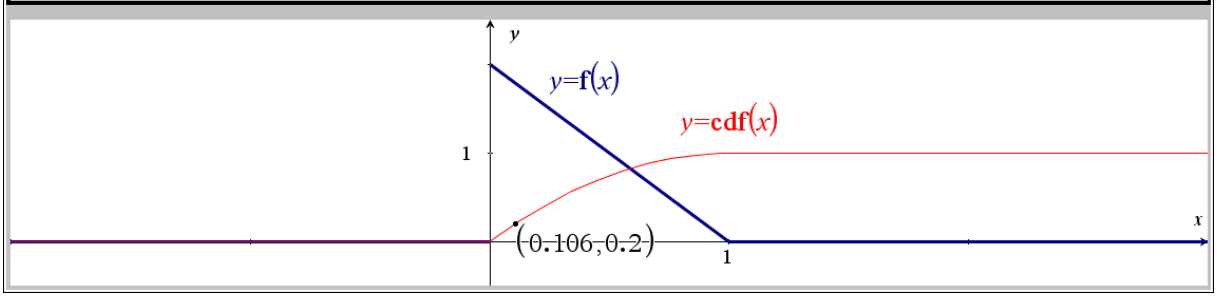
De **cumulatieve verdelingsfunctie** is **F(x)=cdf(x)=P(X≤x) , x**$\in\mathbb{R}$

$\int_0^x (-2\cdot t+2)\,dt \blacktriangleright -x\cdot(x-2)$ $\quad$ $\mathbf{cdf}(x):=\begin{cases}0, & x<0\\ -x^2+2\cdot x, & 0\le x\le 1\\ 1, & x>1\end{cases}$

$\text{solve}(y=-x^2+2\cdot x,x) \blacktriangleright x=-(\sqrt{1-y}-1)$ or $x=\sqrt{1-y}+1$

$F^{-1}(y)=\mathbf{invcdf}(y):=1-\sqrt{1-y}$ with $0\le y\le 1$

$\mathbf{invcdf}(\{0.2,0.3\}) \blacktriangleright \{0.105573,0.16334\}$



Genereer een lijst $\{y_1, y_2, y_3, \ldots, y_n\}$ van *n* data uit de uniforme verdeling op het interval [0,1]

stel $\mathbf{n}:=1000$

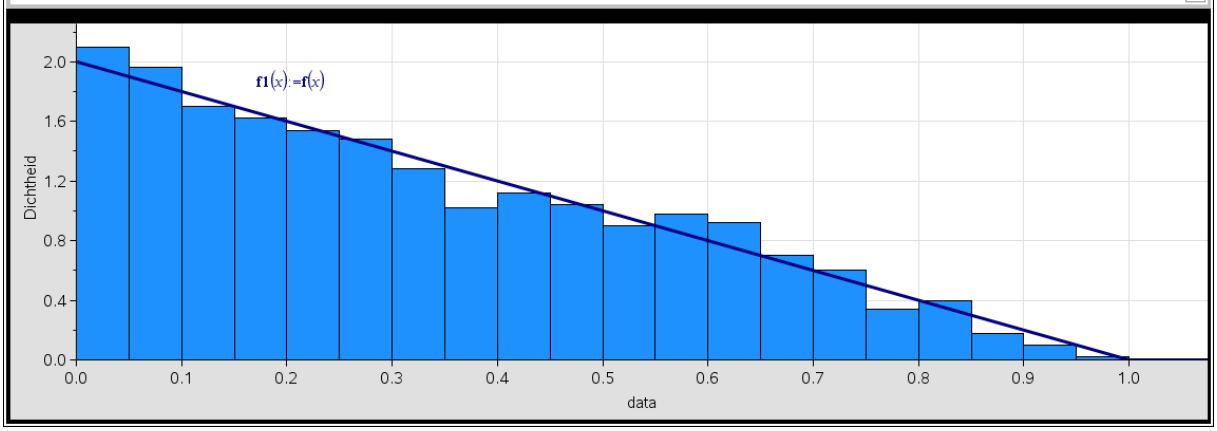
$\mathbf{uniform}:=\text{rand}(\mathbf{n})$

$\blacktriangleright \{0.303912,0.35625,0.838803,0.851539,0.222104,0.033537,0.821229,0.373709,0.898425,0.96073$

De lijst $\{F^{-1}(y_1), F^{-1}(y_2), F^{-1}(y_3), \ldots, F^{-1}(y_n)\}$ levert een steekproef van de gegeven verdeling met dichtheidsfunctie f(x).

$\mathbf{data}:=\mathbf{invcdf}(\mathbf{uniform})$

$\blacktriangleright \{0.165681,0.19766,0.598506,0.614694,0.118016,0.016912,0.577187,0.208614,0.681292,0.80184$



Voor een grote steekproef (bv. $n=1000$) zal het histogram op dichtheidsschaal (met een geschikte klassenbreedte) goed aansluiten bij de grafiek van de dichtheidsfunctie.

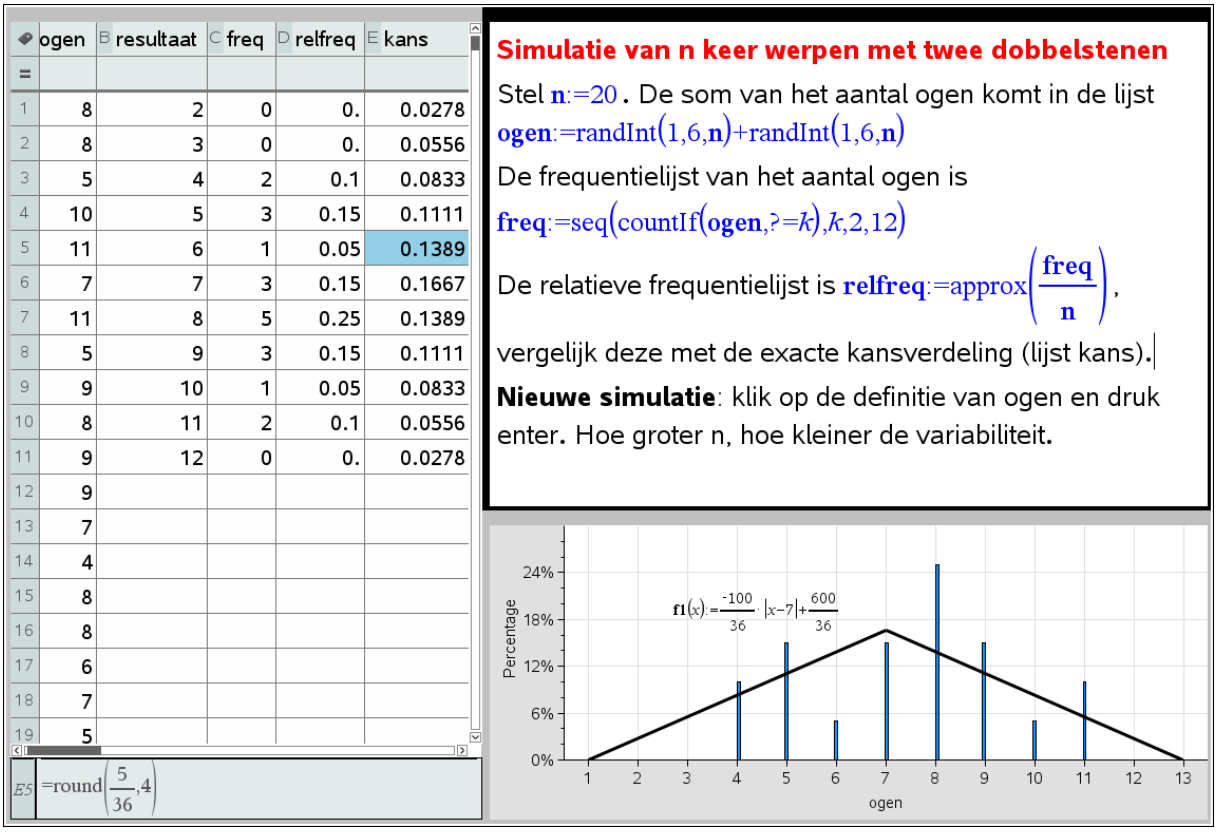
2 Dobbelstenen werpen

2.1 Twee dobbelstenen werpen

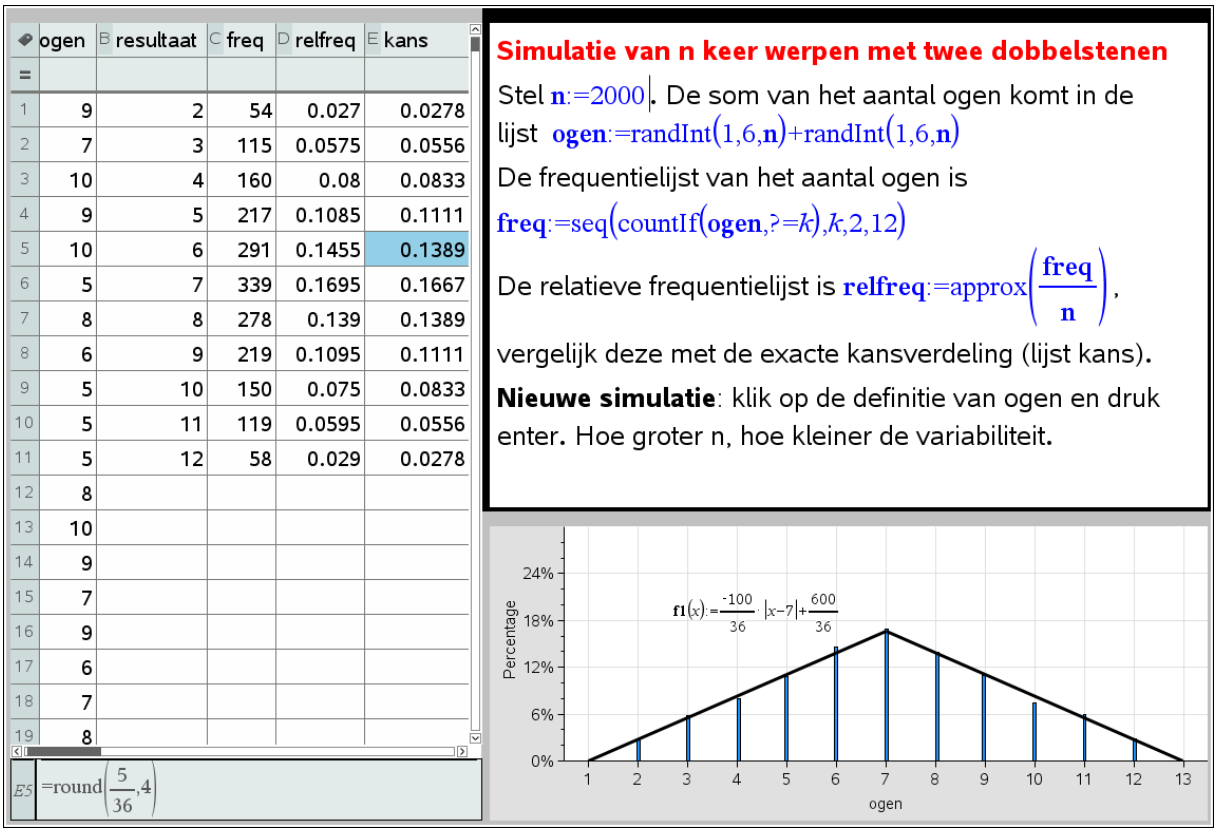
De pagina wordt opgesplitst in drie toepassingen: lijsten en spreadsheet, notities, gegevensverwerking en statistiek. De steekproefgrootte *n* en de simulatie *ogen* worden gedefinieerd bij de notities.

De spreadsheet en het staafdiagram wijzigen dynamisch mee bij vernieuwing van de simulatie en bij keuze van een andere waarde voor *n*.

Onderzoek de invloed van de steekproefgrootte op de steekproefvariabiliteit en de "convergentie" van de relatieve frequenties naar het kansmodel voor twee dobbelstenen.



Simulatie met steekproefgrootte $n = 20$. Grote variabiliteit in de relatieve frequenties van steekproef tot steekproef, meestal een slechte benadering van de exacte kansen.



	ogen	resultaat	freq	relfreq	kans
1	9	2	54	0.027	0.0278
2	7	3	115	0.0575	0.0556
3	10	4	160	0.08	0.0833
4	9	5	217	0.1085	0.1111
5	10	6	291	0.1455	0.1389
6	5	7	339	0.1695	0.1667
7	8	8	278	0.139	0.1389
8	6	9	219	0.1095	0.1111
9	5	10	150	0.075	0.0833
10	5	11	119	0.0595	0.0556
11	5	12	58	0.029	0.0278
12	8				
13	10				
14	9				
15	7				
16	9				
17	6				
18	7				
19	8				

E5 =round(5/36,4)

Simulatie met steekproefgrootte $n = 2000$. Weinig variabiliteit in de relatieve frequenties van steekproef tot steekproef, goede benadering van de exacte kansen.

2.2 Meerdere dobbelstenen werpen

Werp 10 dobbelstenen. Om de exacte kansverdeling te bepalen van het totaal aantal ogen kun je in principe de som berekenen voor elk van $6^{10} = 60466176$ mogelijke uitkomsten (dit zijn geordende 10-tallen), elke uitkomst heeft kans $1/6^{10}$. De frequentietabel van het totaal aantal ogen delen door 6^{10} levert dan de exacte kansverdeling. Deze werkwijze kun je programmeren maar de rekentijd is veel te groot.

Een recursieve aanpak is veel sneller, waarbij we de frequentietabel voor *n* dobbelstenen bepalen uit de frequentietabel voor (*n*-1) dobbelstenen, startend met de tabel voor *n*=1.
Hier volgt de werkwijze voor *n*=1,2,3,4

Som:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
n=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
n=2:	0	1	2	3	4	5	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
n=3:	0	0	1	3	**6**	**10**	**15**	**21**	**25**	**27**	27	25	21	15	10	**6**	**3**	**1**	**0**	**0**	**0**	0	0	0
n=4:	0	0	0	1	4	10	20	35	56	80	**104**	125	140	146	140	125	104	80	56	35	20	**10**	4	1

Merk op dat 104 = 6+10+15+21+25+27, elk getal in een rij is de som van de 6 getallen links ervan in de rij erboven (vul denkbeeldig aan met nullen als je te weinig getallen hebt).
Waarom is dit zo?

Hier volgt het programma:

```
* somfreq                                                    20/20
Define somfreq(aantal)=
Func
© somfreq(aantal) levert een lijst met de frequenties van het totaal aantal ogen bij het werpen van"aantal "
  dobbelstenen.
Local oud,nieuw,n,k,j,i
n:=6· aantal
oud:=seq(0,k,1,n)
nieuw:=oud
For k,1,6
  oud[k]:=1
EndFor
For k,2,aantal
    For j,1,6· k
       nieuw[j]:=0
            For i,1,6
              If j−i>0 Then
                nieuw[j]:=nieuw[j]+oud[j−i]
               EndIf
             EndFor
      EndFor
oud:=nieuw
EndFor
Return right(oud,n−aantal+1)
EndFunc
```

$somfreq(4)$	$\{1,4,10,20,35,56,80,104,125,140,146,140,125,104,80,56,35,20,10,4,1\}$
$somfreq(10)$	$\{1,10,55,220,715,2002,4995,11340,23760,46420,85228,147940,243925,383470,5765$
$\dim(somfreq(10))$	51
$\mathrm{sum}(somfreq(10))$	60466176
6^{10}	60466176

Hiermee kunnen we nu ook dynamisch de kansverdeling voorstellen van het totaal aantal ogen Y bij het werpen van een aantal dobbelstenen n naar keuze. Voor de toevalsvariabele Y geldt $Y = X_1 + X_2 + X_3 + \ldots + X_n$ met X_i = aantal ogen bij de i-de dobbelsteen.

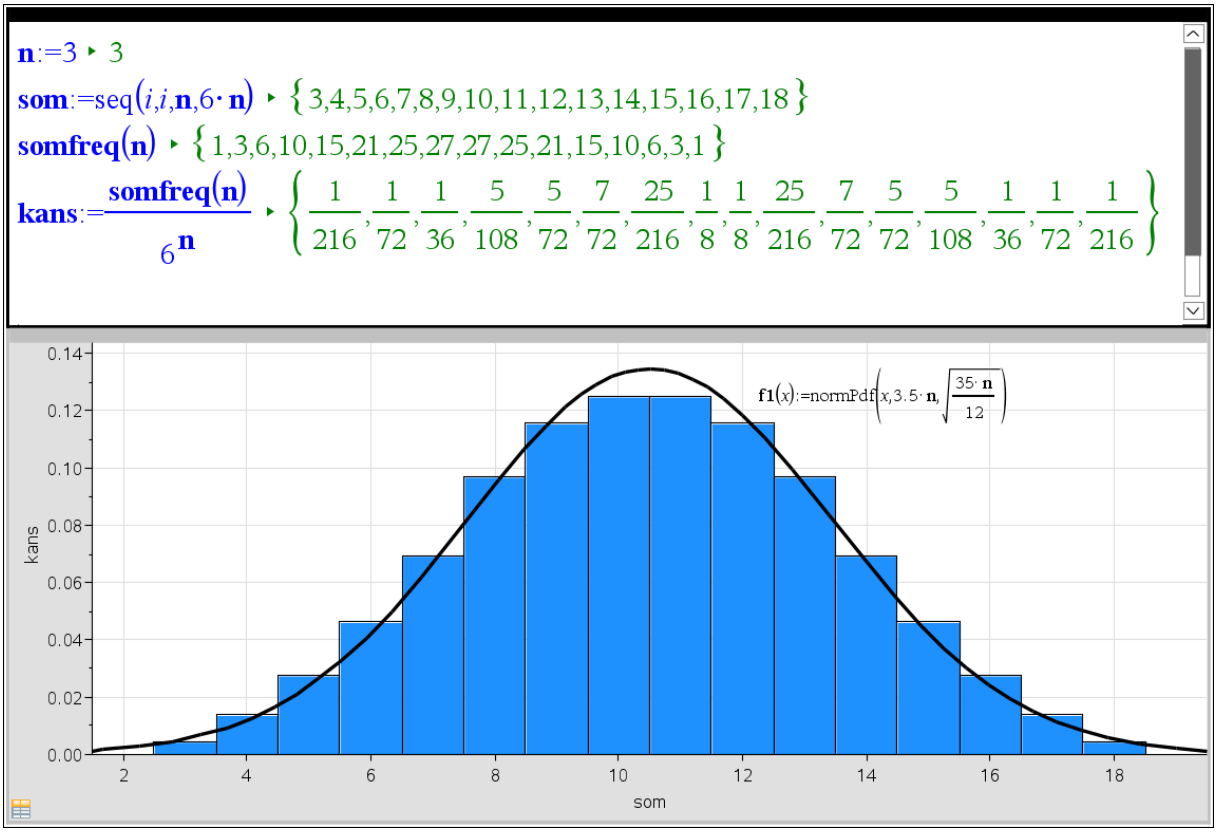
Ga na dat $E(X_i) = 3{,}5$ en $Var(X_i) = \frac{35}{12}$, zodat

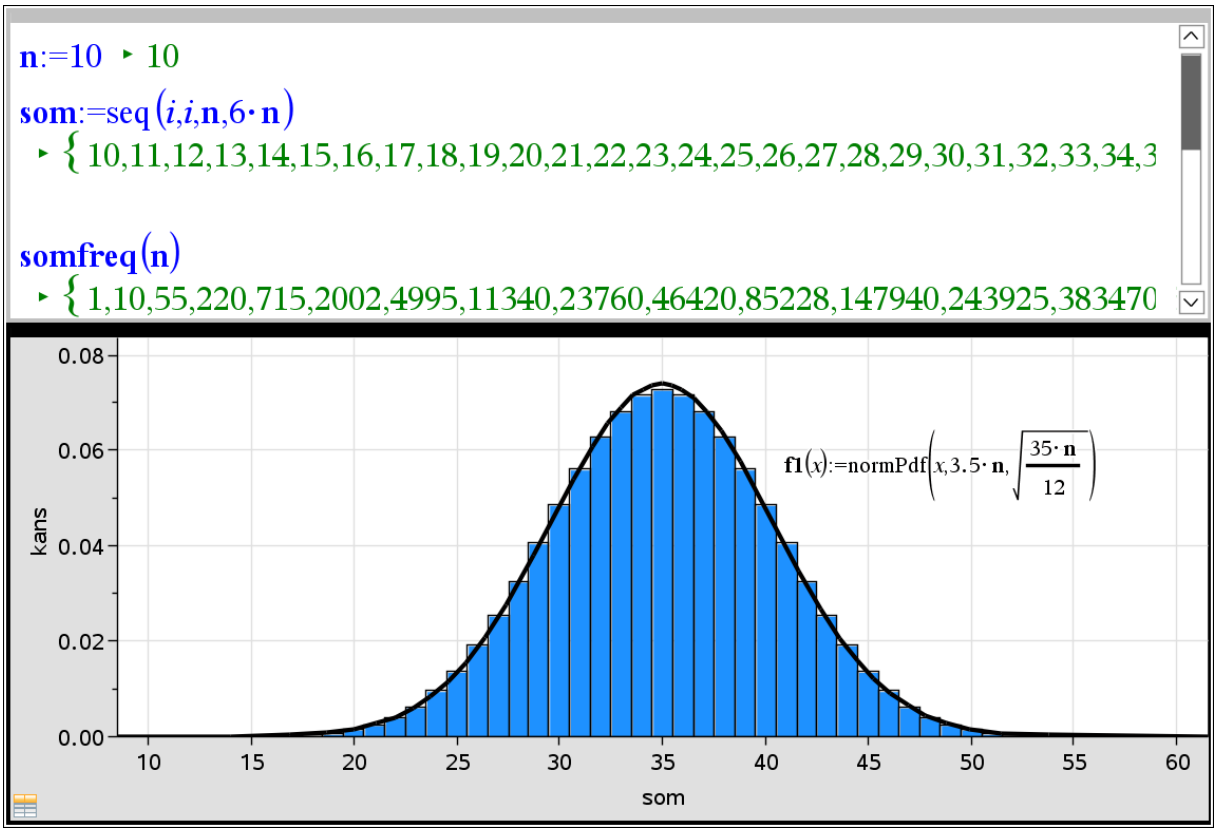
$E(Y) = E(X_1) + E(X_2) + E(X_3) + \ldots + E(X_n) = 3{,}5 \cdot n$ en

$Var(Y) = Var(X_1) + Var(X_2) + Var(X_3) + \ldots + Var(X_n) = \frac{35n}{12}$

Omwille van de centrale limietstelling zal Y nagenoeg normaal verdeeld zijn voor voldoend

grote *n*, het kanshistogram (op dichtheidsschaal met klassenbreedte 1) voor n=10 illustreert dit al duidelijk.





3 Muntstukken werpen

Fischbein en Schnarch (1997) stelden de volgende vraag aan leerlingen:

De kans op minstens 2 keer kop bij 3 keer werpen van een muntstuk is
Kleiner dan / Gelijk aan / Groter dan
de kans op minstens 200 keer kop bij 300 keer werpen van een muntstuk

Laat ons beide situaties 20 keer simuleren (herhaal dit ook eens):

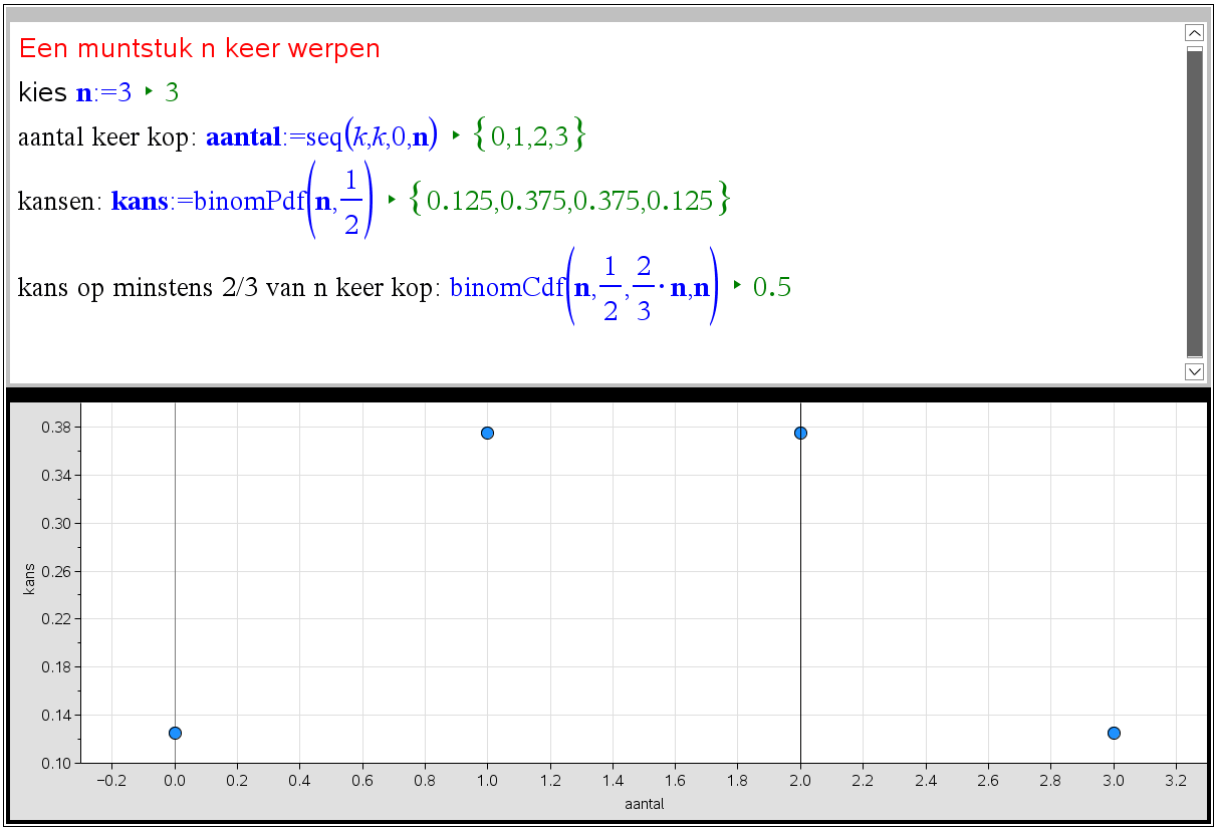
randbin(3,0.5,20) ▸ {2,2,1,2,3,1,1,2,3,1,2,2,1,1,2,1,2,1,2,1}
randbin(300,0.5,20)
▸ {153,146,139,146,130,137,158,142,141,163,143,135,162,151,148,148,145,145,143,173 }

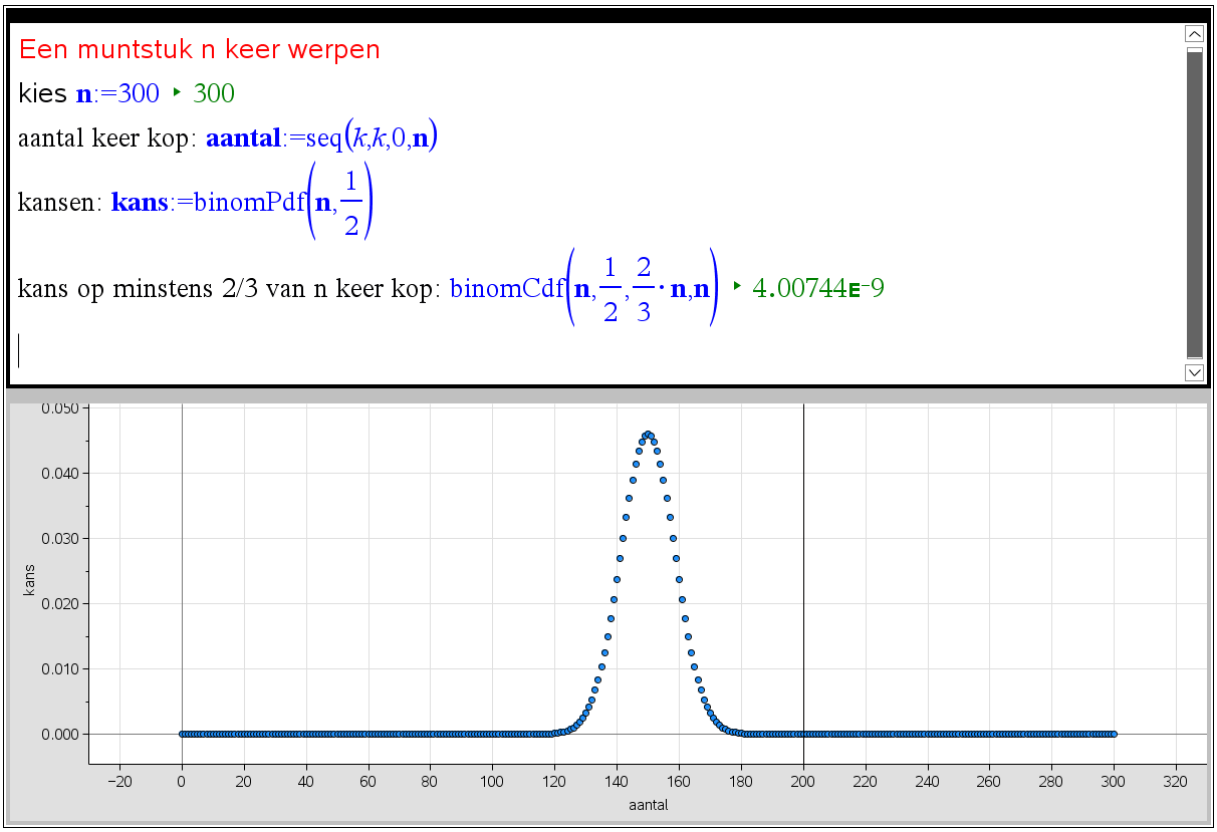
Het antwoord op de vraag blijkt duidelijk “Groter dan”.

Het aantal leerlingen met het verkeerde antwoord “Gelijk aan” (meest voorkomende fout) nam toe met de leeftijd van de leerlingen van 30% (leeftijd 11 jaar) tot 80% (leeftijd 17 jaar)!

Het antwoord is ook zonder simuleren evident als we denken aan de wet der grote aantallen: bij 300 keer werpen van een munstuk verwachten we ongeveer 150 keer kop, aangezien de relatieve frequentie hier al een goede benadering zal leveren voor de kans op kop 1/2.

Het is ook mogelijk om de kansen te berekenen in beide situaties:





4 Kansverdelingen ontdekken

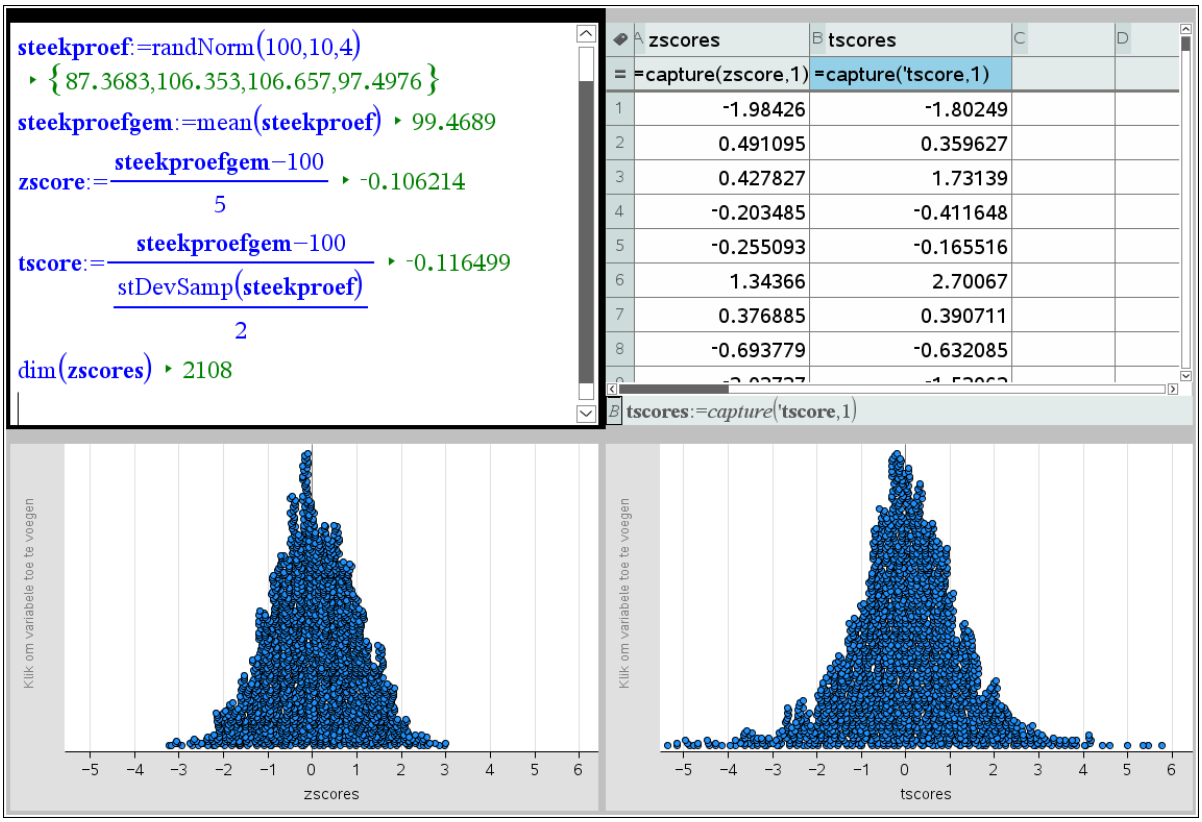
4.1 z-scores versus t-scores

Stel $X \sim N(\mu,\sigma)$ (de toevalsvariabele X is normaal verdeeld met gemiddelde μ en standaardafwijking σ).
Trek uit deze populatie een steekproef met grootte n, dan geldt voor het steekproefgemiddelde: $\overline{X} \sim N\left(\mu,\frac{\sigma}{\sqrt{n}}\right)$.

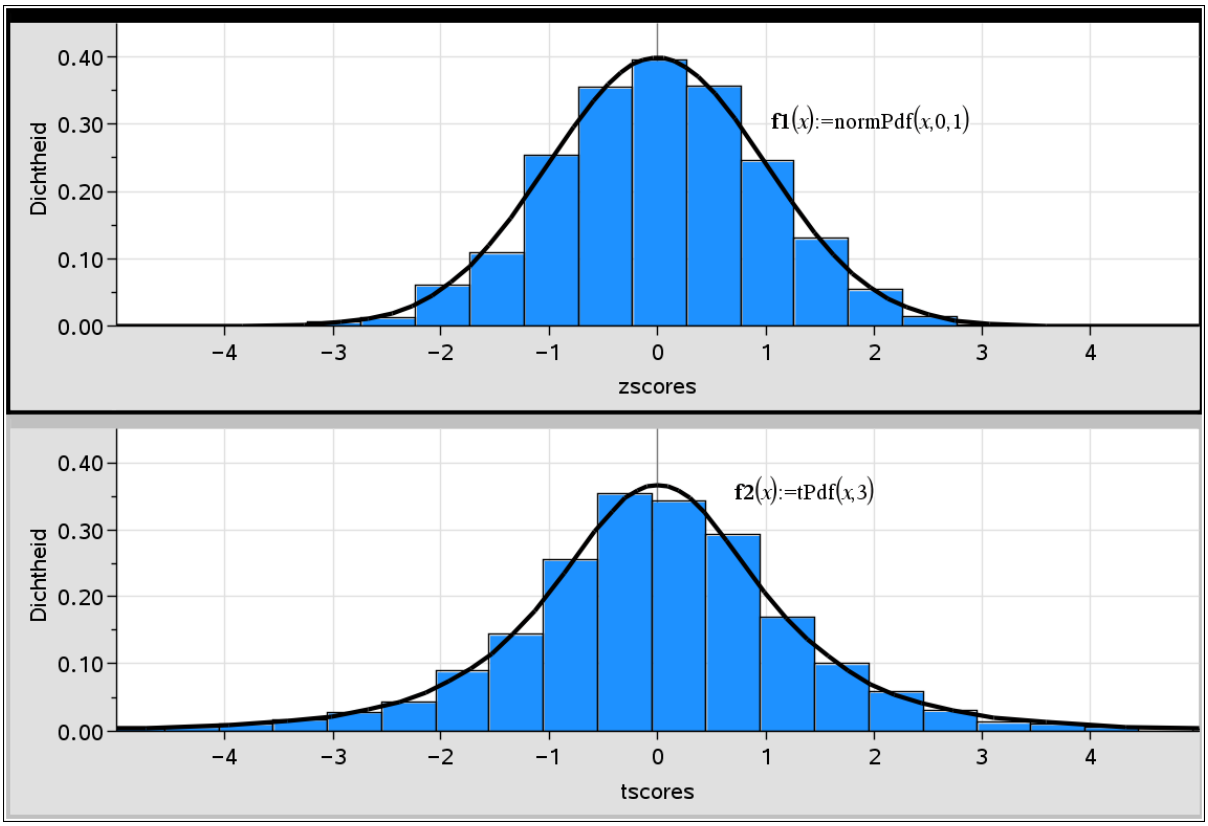
Standaardiseren levert $Z = \dfrac{\overline{X}-\mu}{\frac{\sigma}{\sqrt{n}}} \sim N(0,1)$.

Als de populatiestandaardafwijking σ niet gekend is, dan wordt ze geschat met de steekproefstandaardafwijking s. Zo ontstaat de nieuwe toevalsvariabele $T = \dfrac{\overline{X}-\mu}{\frac{S}{\sqrt{n}}}$.

We vergelijken de "z-scores" $z = \dfrac{\overline{x}-\mu}{\frac{\sigma}{\sqrt{n}}}$ met de "t-scores" $t = \dfrac{\overline{x}-\mu}{\frac{s}{\sqrt{n}}}$ via hun dichtheidshistogrammen in een concrete simulatie met populatiegemiddelde $\mu = 100$, populatiestandaardafwijking $\sigma = 10$, steekproefgrootte $n = 4$.



De t-scores vertonen een grotere spreiding dan de z-scores.



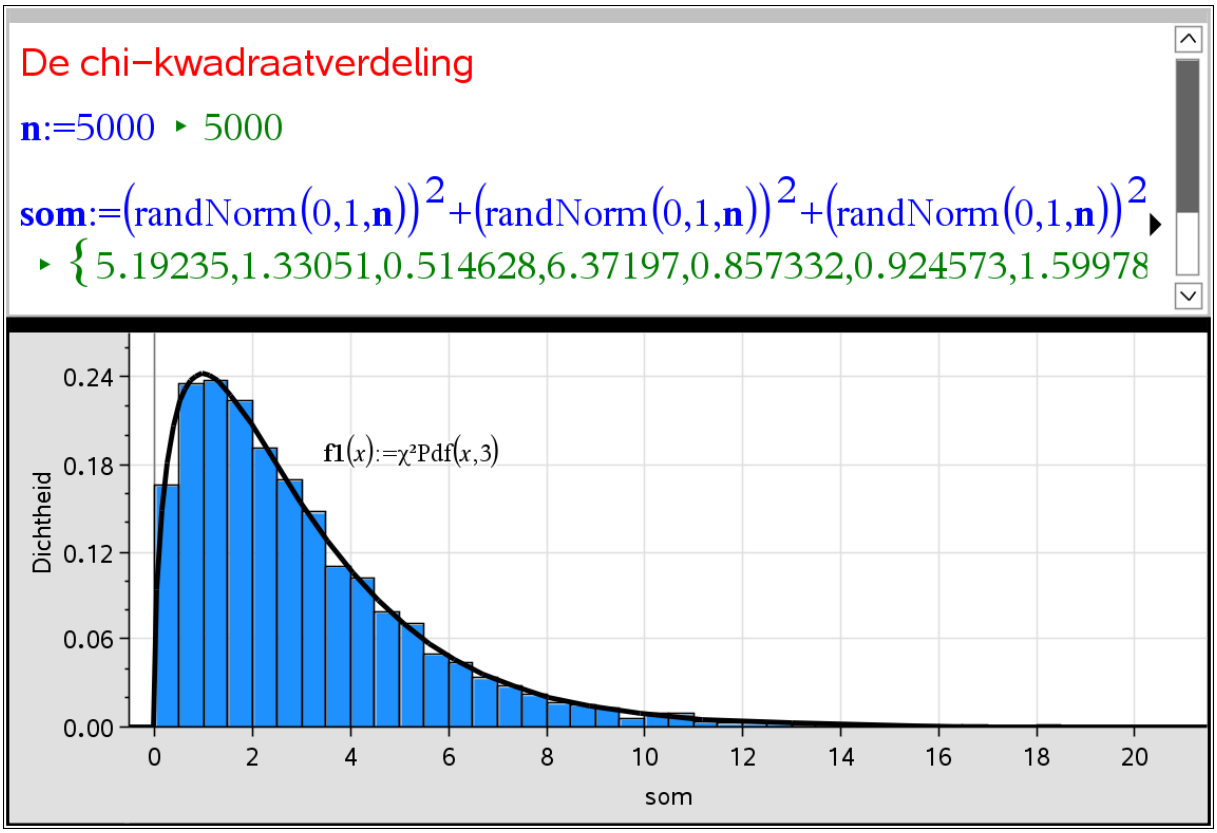
De t-scores zijn verdeeld volgens een t-verdeling met 3 vrijheidsgraden ($3 = n - 1$), wel klokvormig maar niet normaal verdeeld!

4.2 De chi-kwadraatverdeling

Stel dat Z_1, Z_2, Z_3 onafhankelijke standaardnormaal verdeelde toevalsvariabelen zijn: $Z_i \sim N(0,1)$.

Wat is de kansverdeling van de toevalsvariabele $X = Z_1^2 + Z_2^2 + Z_3^2$?

Werkwijze: kies lukraak drie getallen uit een standaardnormale verdeling , kwadrateer die getallen en tel ze op, dit levert alvast een aangenomen waarde $x = z_1^2 + z_2^2 + z_3^2$ van de toevalsvariabele *X*. De waarde van *x* is hierbij onvoorspelbaar en de getallen z_1, z_2, z_3 komen naar ons toe volgens het kansmechanisme van een standaardnormale dichtheidsfunctie.
Doe dit heel vaak en zet de zo verkregen getallen *x* uit in een histogram op dichtheidsschaal, de wet der grote aantallen garandeert dat het histogram goed zal aansluiten bij de exacte kansverdeling (dichtheidsfunctie) van *X*.



De variabele *X* heeft een chi-kwadraatverdeling met drie vrijheidsgraden.

Onderzoek nu de kansverdeling van $Y = Z_1^2 + Z_2^2 + Z_3^2 + Z_4^2$, met $Z_i \sim N(0,1)$.

5 De gewetensvolle professor

Uit ervaring weet een professor dat de examenscores van zijn grote groep studenten normaal verdeeld zijn met een gemiddelde van 11 op 20 en standaardafwijking 2,5 : $\mu = 11$ en $\sigma = 2{,}5$.

De gewetensvolle professor stelt volgens hem vragen van dezelfde moeilijkheidsgraad aan groepen van 15 studenten (examens in de voormiddag en namiddag).

In één groep zijn er maar 5 studenten geslaagd, in een andere groep zijn er 10 geslaagd. Moet de prof zich zorgen maken over de moeilijkheidsgraad van de examens?

Laat ons dit eerst eens simuleren:

```
groep1:=round(randNorm(11,2.5,15),0) ▸ {12.,14.,12.,12.,14.,14.,14.,12.,13.,10.,17.,8.,11.,11.,9.}
groep2:=round(randNorm(11,2.5,15),0) ▸ {12.,10.,10.,15.,10.,15.,12.,10.,16.,11.,7.,6.,11.,9.,11.}
groep3:=round(randNorm(11,2.5,15),0) ▸ {13.,11.,10.,12.,10.,14.,12.,11.,13.,9.,12.,5.,8.,10.,5.}
groep4:=round(randNorm(11,2.5,15),0) ▸ {9.,12.,9.,9.,7.,13.,11.,13.,10.,11.,7.,19.,7.,8.,12.}
groep5:=round(randNorm(11,2.5,15),0) ▸ {14.,8.,9.,13.,9.,13.,11.,10.,10.,14.,10.,9.,6.,14.,14.}
countIf(groep1,?≥10) ▸ 13
countIf(groep2,?≥10) ▸ 12
countIf(groep3,?≥10) ▸ 11
countIf(groep4,?≥10) ▸ 8
countIf(groep5,?≥10) ▸ 10
```

Simulaties van 5 groepen levert minstens 8 studenten geslaagd, maar misschien is dit toeval ...

Hier volgt een simulatie van 30 groepen:

```
geslaagd:=seq(countIf(randNorm(11,2.5,15),?≥9.5),k,1,30)
 ▸ {13,10,10,12,10,13,12,9,13,10,10,11,12,9,11,11,8,10,12,10,10,11,10,10,11,10,11,9,11,12}
```

Ook voor deze 30 groepen zijn er telkens minstens 8 studenten geslaagd.
Nu begint de prof toch ongerust te worden…

De volgende pagina toont een simulatie van 2000 groepen, er is maar één groep met 5 studenten geslaagd…

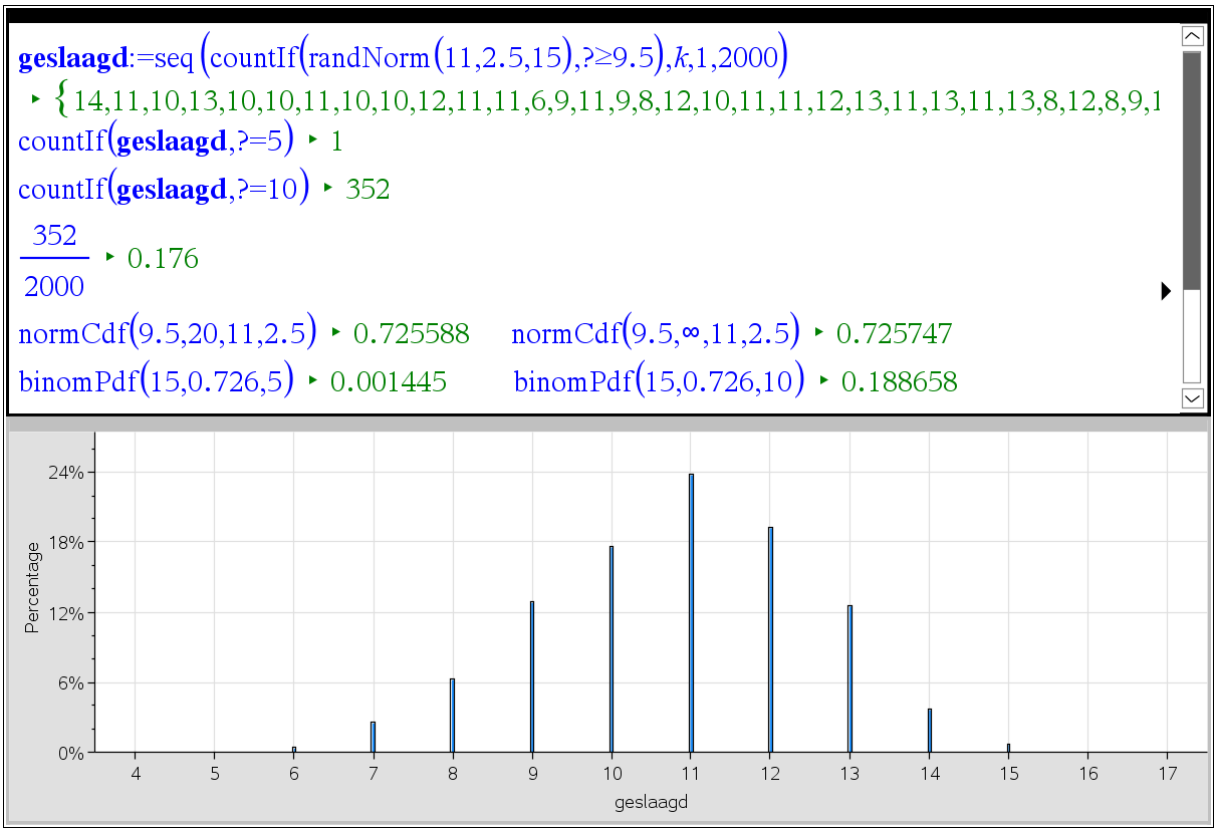
De prof beseft dat er iets aan de hand is en berekent de exacte kans. De kans dat een student slaagt is 0,726 :

```
normCdf(9.5,20,11,2.5) ▸ 0.725588
```

Het aantal studenten dat slaagt voor een groep van 15 studenten is binomiaal verdeeld met parameters n = 15 en p = 0,726.
De kans dat er 5 studenten geslaagd zijn is slechts 0,14%:

```
binomPdf(15,0.726,5) ▸ 0.001445
```

De prof zal dus best de examenvragen nog eens onderzoeken.



Merk op dat de kans op 10 studenten geslaagd 18,9% is, deze simulatie van 2000 groepen levert een relatieve frequentie van 17,6% als benadering.