Kleine populaties

Wouter Stekelenburg, Albert-Jan Yzelman

20-05-2003

1 Inleiding

In dit verslag gaat het erom, hoe we de populatieverandering van een of andere populatie kunnen simuleren. We gaan het hier hebben over een populatie neushoorns, 96 in totaal. Een onderzoek over neushoorns in het algemeen toont aan dat de dieren globaal over 5 leeftijdsklassen verdeeld kunnen worden. Verder heeft men de kansen kunnen berekenen dat een neushoorn overleeft van een leeftijdsklasse tot de volgende, en de kans dat een paar neushoorns uit een bepaalde klasse jongen krijgt. Deze kansen staan hieronder samengevat:

Klasse	1	2	3	4	5
Overleven	3/5	2/3	3/4	1/3	0
Voortplanting	0	1/6	5/4	1	1

De verdeling van onze populatie van 96 over de leeftijdsklassen is als volgt:

Klasse	1	2	3	4	5
Aantal	40	24	16	12	4

Nu is het zaak om, uit bovenstaande informatie, een redelijk efficiente methode te vinden om te simuleren hoe deze populatie zich gedraagt. Dit gaan wij doen met behulp van zogenaamde lesliematrices.

2 Het model

2.1 Lesliematrices

Een lesliematrix is een n x n matrix die bijzonder goed te gebruiken is bij dit soort populatiemodellen. De standaardvorm voor een Lesliematrix is:

$$\begin{pmatrix} g_1 & g_2 & \cdots & g_{n-1} & g_n \\ s_1 & 0 & \ldots & 0 & 0 \\ 0 & s_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & s_{n-1} & 0 \end{pmatrix}$$

Hier is n gelijk aan de hoeveelheid leeftijdsklassen. g staat voor de growth, de kans op geboortes dus. De s staat voor survival, dus hoeveel neushoorns er gemiddeld blijven overleven (naar de volgende klasse). Het is niet moeilijk in te zien dat als zo'n lesliematrix wordt vermenigvuldigd met een 1 x n matrix, waarin een populatie is weergegeven, een 1 x n matrix wordt teruggegeven waarin de verwachtte (gemiddelde) volgende populatie staat. In ons geval valt er voor onze neushoornpopulatie dus ook zo'n lesliematrix op te stellen, en deze wordt dan:

$$\begin{pmatrix} 0 & 1/6 & 5/4 & 1 & 1 \\ 3/5 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 \end{pmatrix}$$

Nu kunnen we dus al berekenen wat de verwachtte grootte van de volgende populatie zal worden:

$$\begin{pmatrix} 0 & 1/6 & 5/4 & 1 & 1 \\ 3/5 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 \end{pmatrix} . \begin{pmatrix} 40 \\ 24 \\ 16 \\ 12 \\ 4 \end{pmatrix} = \begin{pmatrix} 40 \\ 24 \\ 16 \\ 12 \\ 4 \end{pmatrix}$$

Volgens onze matrix verandert de populatie gemiddeld dus niet. Waarom is dat zo?

2.2 Eigenvectoren en dominante eigenwaardes

Om beter te begrijpen waarom, moeten we onze lesliematrix wat beter bekijken. Laten we deze in vervolg matrix A noemen. Zoals we weten, hangt het gedrag van een transformatie veelal af van eigenvectoren en eigenwaardes. We kunnen mbv Mathematica eigenwaardes berekenen: Dit doen we dmv het commando:

```
N[Abs[Eigenvalues[A]]]
```

Het antwoord is als volgt:

```
{1., 0.455132, 0.694804, 0.455132, 0.694804}
```

We zien dus, dat de matrix A een dominante eigenwaarde heeft, welke ook nog eens gelijk is aan 1. We zoeken ook de bijbehorende eigenvector E_1, en deze is gelijk aan $(10, 6, 4, 3, 1)^t$. Uit de theorie is dan bekend (vanwege $\lambda_1 = 1$) dat:
$A.E_1 = \lambda_1.E_1$
en dus in het bijzonder voor ons geval dat:

$$\begin{pmatrix} 10 \\ 6 \\ 4 \\ 3 \\ 1 \end{pmatrix} . \begin{pmatrix} 0 & 1/6 & 5/4 & 1 & 1 \\ 3/5 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 6 & 4 & 3 & 1 \end{pmatrix}$$

Nu merken we op dat onze beginpopulatie, $(40, 24, 16, 12, 4)^t$ een viervoud is van E_1. Dus, bovenstaande gelijkheid geldt ook als we E_1 vervangen door onze beginpopulatie. Hiermee is dus verklaard waarom we verwachten dat het populatie-gemiddelde gelijk blijft na een transformatie. Dit betekent dus dat onze neushoornpopulatie een zeer stabiele populatie is. Verdere simulatie zal dit ook uitwijzen.

2.3 Modelleren

Om een echte simulatie uit te voeren is het natuurlijk niet handig om telkens bovenstaande matrixvermenigvuldiging toe te passen. Immers, we krijgen dan alleen maar gemiddeldes eruit, en we weten al dat die telkens hetzelfde zijn. Wat wij willen is dat er met behulp van Randomfuncties (kansfuncties) een andere populatie wordt uitgerekend. Een stochastische bewerking dus. Als er bijvoorbeeld 40 eerste-klassers zijn, waarvan er dus 80

```
survive[kans_] := If[Random[] < kans, 1, 0]
```

Hier geldt dus dat als het random-getal kleiner is dan de kans, dat de functie 1 teruggeeft (ofwel, de neushoorn overleeft) en anders 0 (de neushoorn is dood). Als we nu deze functie meerdere keren achterelkaar toepassen, zoals:

```
Sum[survive[3/5], {40}]
```

Dit levert precies hetgeen op dat we wilden; het aantal neushoorns uit klasse 1 die overleven, stochastisch berekend. Voor de geboortes moeten we ook nog een functie hebben:

```
poisson[gemiddeld_] :=
 If[gemiddeld > 0, Random[PoissonDistribution[gemiddeld]], 0];
```

Deze functie geeft per (vrouwtjes-) neushoorn hoeveel jongen deze krijgt. Ook deze functie wordt gebruikt als in het volgende voorbeeld:

```
Sum[poisson[1/6], {24}]
```

(Dit is dus de berekening uitgevoerd om te kijken hoeveel jongeren er uit leeftijdsklasse 2 worden geboren)

2.4 De module

Welnu, met behulp van de twee bovenstaande functies, kunnen we in Mathematica een module defenieren, die volledig automatisch uit een beginpopulatie en een lesliematrix de volgende populatie berekent. Zo'n moduul is als volgt geschreven:

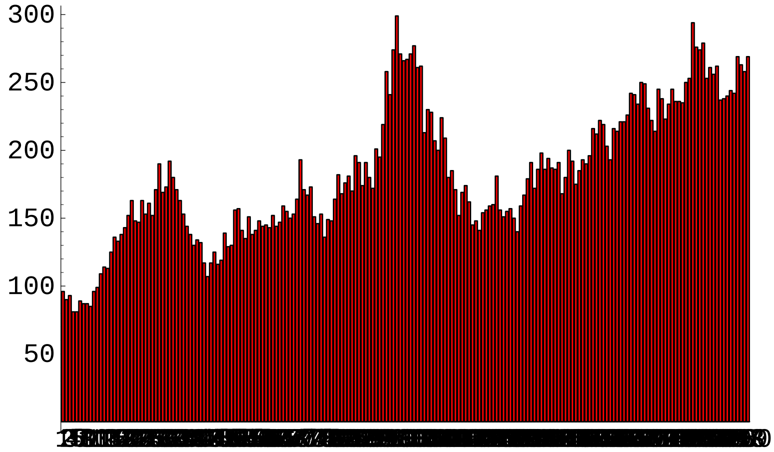
```
1 lesliesim[huidigePopulatie_, lesliematrix_] :=
2  Module[{volgendePopulatie, i, j, maxLeeftijd = Length[lesliematrix]},
3   volgendePopulatie = 0*huidigePopulatie;
4   Do[volgendePopulatie[[1]] =
6    volgendePopulatie[[1]] +
7    Sum[poisson[lesliematrix[[1, i]]], {huidigePopulatie[[i]]}], {i,
8    maxLeeftijd}];
9   Do[volgendePopulatie[[i]] =
10   Sum[survive[
11   lesliematrix[[i, i - 1]]], {huidigePopulatie[[i - 1]]}], {i, 2,
12   maxLeeftijd}];
13   volgendePopulatie
14  ]
```

Regel 1 defeniert een functie lesliesim afhankelijk van huidigepopulatie en lesliematrix. Regel 2 duidt het begin van de module aan. Tevens staan er tussen haakjes de locale variabelen. Regel 3 maakt van volgendepopulatie een 0-vector ter grootte van huidigepopulatie. Regel 4-8 berekenen stochastisch hoeveel neushoorns geboren worden uit de huideige populatie, en geven dat getal door aan volgendepopulatie. Regel 9-12 berekenen stochastisch hoeveel neushoorns er overleven van een leeftijd tot de volgende, en geven de getallen door aan volgendepopulatie. Regel 13 laat lesliesim de volgendepopulatie retourneren, en regel 14 sluit de module af.

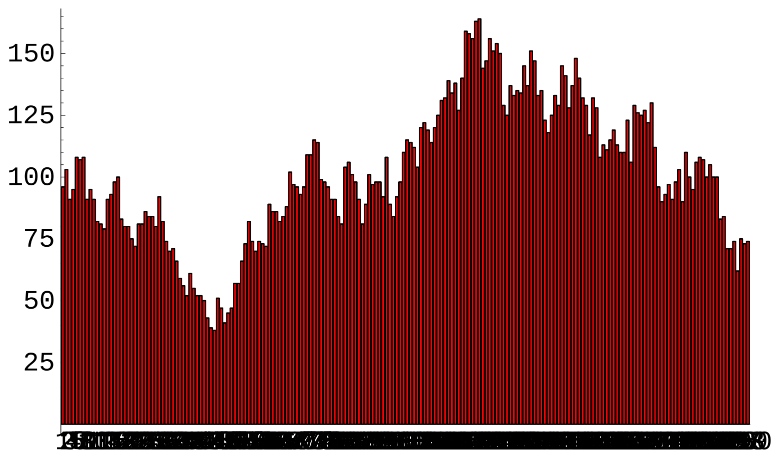
Met behulp van bovenstaande functie, kunnen we weer een andere functie defeniren die een mogelijke verloop van een neushoornpopulatie simuleert, en weergeeft in een mooie grafiek. De functie die we hiervoor hebben gemaakt heet jopi, en de definitie staat hieronder:

```
jopi[startpop_, lesli_, generaties_] :=
 Module[{his},
 For[pop = startpop; his = {}; i = 1, i <= generaties,
  pop = lesliesim[pop, lesli]; i++, his = Append[his, pop]];
 BarChart[Apply[Plus, his, 1]]
 ]
```

Deze module laat lesliesim i keer lopen, waar i gelijk is aan het getal dat bij generaties is opgegeven. Elke generatie wordt de populatie opgeslagen in de verzameling his. Deze wordt later geplot, met behulp van een barchart. Een voorbeeldresultaat is bijvoorbeeld:



Deze is gegenereerd met onze standaard begin populatie en onze lesliematrix A. Voor generaties is 200 opgegeven. Verschillende keren dezelfde functie laten lopen (dus met dezelfde waardes) hoeven absoluut geen overeenkomstige resultaten op te leveren, immers, er is gebruik gemaakt van stochastische functies. Het volgende plaatje illustreert dit.



Na velen malen proberen zien we dat het werkelijk veel kanten op kan gaan met een neushoornpopulatie. Echter blijkt altijd dat de bevolking over lange termijn redelijk stabiel is. De grootte van onze beginpopulatie is dus wel groot genoeg om dit lesliemodel goed te gebruiken.

Een gemiddelde afwijking in populatie-aantal is te berekenen door de som te nemen van alle relevante varianties in dit model. Dus stel ons model heeft als beginpopulatie: $(a, b, c, d, e)^t$ Volgens matrix A, is de kans dat een neushoorn van klasse 1 overleeft naar klasse 2, 80 procent. De variantie van deze overlevering, rekening houdend met de beginpopulatie, is dan gelijk aan $\frac{3\sqrt{a}}{5}$. Voor de klasse-overleving van 2 naar 3 is de variantie dan $\frac{2\sqrt{b}}{3}$, enzovoorts. Niet is te vergeten dat er ook nog varianties moeten worden uitgerekend voor de geboortes van neushoorns. De som van deze varianties is de variantie van de totale bevolkingsgrootte.
Als men deze redenering volgt, komt je na wat rekenwerk uit op de volgende formule: $\frac{3\sqrt{a}}{5} + \frac{5\sqrt{b}}{6} + 2\sqrt{c} + \frac{4\sqrt{d}}{3} + \sqrt{e}$
Als we voor $(a, b, c, d, e)^t$ onze standaard beginpopulatie invoeren, komt er uit bovenstaande formule ongeveer $22, 5$. Dit lijkt ongeveer overeen te komen met wat we in de plaatjes zien. Let wel, deze variantie geldt dus alleen van 1 populatie naar de volgende, dus niet bijvoorbeeld van de 1e populatie naar de 50e, die kunnen wel meer dan dat verschillen. In een functie zoals jopi is het trouwens ook makkelijk een regeltje toe te voegen die het gemiddelde bekijkt van alle resultaten. Men zal zien, dat deze heel erg dichtbij 96 ligt. Een voorbeeld van zo'n uitgebreide jopi-functie is bijvoorbeeld:

```
midjopi[startpop_, lesli_, generaties_] :=
 Module[{his, rij, res},
  For[rij = {}; j = 1, j < 10,
   For[his = {}; i = 1, i <= generaties, his = lesliesim[startpop, lesli];
    i++]; rij = Append[rij, Sqrt[his].Sqrt[his]]; j++];
  BarChart[Apply[Plus, rij, 1]];
  N[Mean[rij]]
 ]
```

Deze functie geeft dan naast de bekende staafgrafiek ook het gemiddelde weer, in cijfers. Veelvuldig uittesten leert inderdaad dat het gemiddelde dicht bij 96 ligt.

2.5 Betrouwbaarheid

Met wat wij als beginpopulatie hebben genomen, lijkt de simulatie goed te gelukken. De neushoorns sterven niet uit, en het gemiddelde blijft rond de 96. Vraag is nu natuurlijk of de simulatie nog steeds lukt met andere beginpopulaties. Bij grotere populaties hoort er niet veel anders te gebeuren als bij de huidige beginpopulatie. In ieder geval zal nog steeds een redelijk stabiele populatie zichtbaar zijn. Gaat men echter veel lager zitten dan 96 neushoorns, dan kan het nogal snel gebeuren dat de hele populatie uitsterft. Als dit niet gebeurt, dan is het vaak zo dat de populatie eerst explosief is gegroeid, om daarna pas een beetje te stabiliseren. Simulatie wijst hier dus uit dat de populatie dan juist zelden stabiel blijft. Het model is hier niet meer betrouwbaar. Alhoewel er best een wiskundige redenering kan worden opgesteld over wanneer het model onbetrouwbaar wordt, hebben we ons toch maar beperkt tot simpel proberen. Het lijkt dat als de beginpopulatie ≥ 80, dat dan de bevolking redelijk stabiel is. Bij minder dan 80 gebeurt het al snel dat de populatie uitsterft, of juist eerst explosief groeit alvorens te stabiliseren.

3 Conclusie

Met wat ons gegeven is vanuit een extern onderzoek over neushoorns, hebben wij getracht een simulatie op te zetten van een neushoornpopulatie met behulp van een relatief simpel model. Relatief simpel, omdat in de natuur natuurlijk wel meer factoren spelen dan geboorte en sterfte, en omdat die 2 factoren natuurlijk niet altijd constant zijn. Desondanks kan dit model verklaren waarom neushoornpopulaties doorgaans redelijk stabiel zijn, zonder daar al te veel factoren bij te halen. Verder hebben we geprobeerd te bekijken hoe betrouwbaar dit model is, en hebben we een soort van minimum-begin-populatie van 80 gevonden, omdat anders de betrouwbaarheid naar onze mening behoorlijk wat tekort schiet.