Prof. Dr. Dieter Riebesehl und Prof. Dr. Dörte Haftendorn 2011

Herleitung, Programmierung, Diskussion

Erzeugung stochastischer Vektoren und Matrizen

Prof. Dr. Dieter Riebesehl+Dörte Haftendorn, Leuphana Universität Lüneburg, Sept. 2011

www.mathematik-verstehen.de

Spielwiese mit den hier bereitgestellten Möglichkeiten:

myv:=stochvector(3) ▸ [0.617002 0.185868 0.19713] round(myv,2) ▸ [0.62 0.19 0.2]

So kann man stochastische Vektoren erzeugen.

mym:=stochmatrix(3) ▸ $\begin{bmatrix}0.106652 & 0.042972 & 0.850376\\0.529917 & 0.296396 & 0.173687\\0.911463 & 0.005741 & 0.082796\end{bmatrix}$ round(mym,2) ▸ $\begin{bmatrix}0.11 & 0.04 & 0.85\\0.53 & 0.3 & 0.17\\0.91 & 0.01 & 0.08\end{bmatrix}$

eine stochastische Matrix mit Zeilensummen 1

myv·mym ▸ [0.343976 0.082736 0.573288] eine Zustandsvektor-Änderung

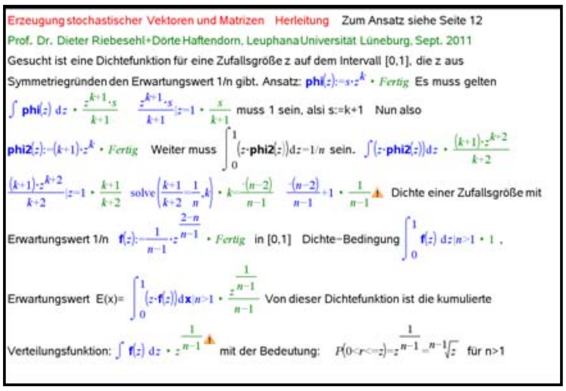
stabil:=mym^{50} ▸ $\begin{bmatrix}0.497789 & 0.03422 & 0.467992\\0.497783 & 0.03422 & 0.467997\\0.49778 & 0.03422 & 0.468\end{bmatrix}$ Die stabile Übergangsmatrix

eig:=stabil[1] ▸ [0.497789 0.03422 0.467992] der Eihgenvektor

eig·mym ▸ [0.497781 0.03422 0.467999] Beweis, dass er Eigenvektor ist.

Dieses kann man verwenden um (viele) Beispiele zu erfinden.

1.1



Erzeugung stochastischer Vektoren und Matrizen Herleitung Zum Ansatz siehe Seite 12

Prof. Dr. Dieter Riebesehl+Dörte Haftendorn, Leuphana Universität Lüneburg, Sept. 2011

Gesucht ist eine Dichtefunktion für eine Zufallsgröße z auf dem Intervall [0,1], die z aus Symmetriegründen den Erwartungswert 1/n gibt. Ansatz: phi(z):=$s\cdot z^k$ ▸ Fertig Es muss gelten

$\int$ phi(z) dz ▸ $\frac{z^{k+1}\cdot s}{k+1}$ $\quad \frac{z^{k+1}\cdot s}{k+1}|z=1$ ▸ $\frac{s}{k+1}$ muss 1 sein, alsi s:=k+1 Nun also

phi2(z):=$(k+1)\cdot z^k$ ▸ Fertig Weiter muss $\int_0^1 (z\cdot \text{phi2}(z))\,dz=1/n$ sein. $\int (z\cdot\text{phi2}(z))\,dz$ ▸ $\frac{(k+1)\cdot z^{k+2}}{k+2}$

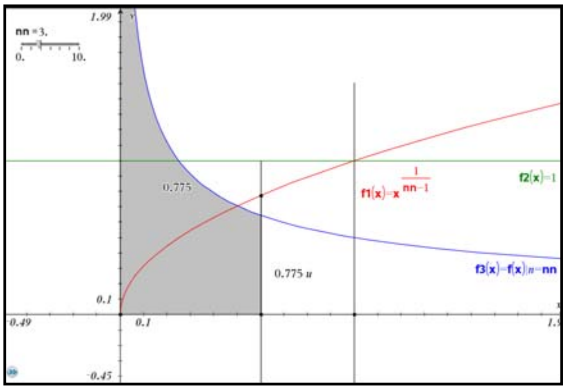
$\frac{(k+1)\cdot z^{k+2}}{k+2}|z=1$ ▸ $\frac{k+1}{k+2}$ solve$\left(\frac{k+1}{k+2}=\frac{1}{n},k\right)$ ▸ $k=\frac{-(n-2)}{n-1}$ $\quad \frac{-(n-2)}{n-1}+1$ ▸ $\frac{1}{n-1}$ Dichte einer Zufallsgröße mit

Erwartungswert 1/n f(z):=$\frac{1}{n-1}\cdot z^{\frac{2-n}{n-1}}$ ▸ Fertig in [0,1] Dichte-Bedingung $\int_0^1$ f(z) dz|n>1 ▸ 1 .

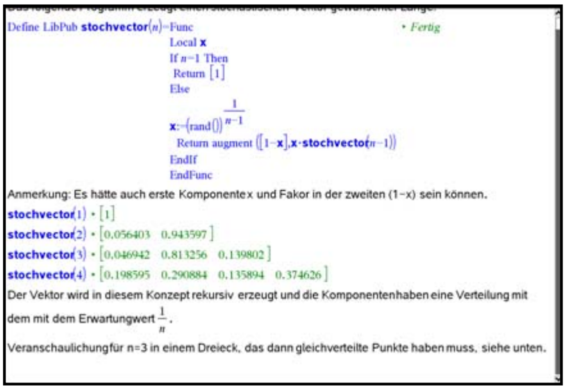
Erwartungswert E(x)= $\int_0^1 (z\cdot f(z))\,dx|n>1$ ▸ $\frac{z^{\frac{1}{n-1}}}{n-1}$ Von dieser Dichtefunktion ist die kumulierte

Verteilungsfunktion: $\int$ f(z) dz ▸ $z^{\frac{1}{n-1}}$ mit der Bedeutung: $P(0<r<=z)=z^{\frac{1}{n-1}}=\sqrt[n-1]{z}$ für n>1

1.2



1.3



Define LibPub stochvector(n)=Func ▸ Fertig

```
Local x
If n=1 Then
 Return [1]
Else
x:=(rand())^(1/(n-1))
 Return augment([1-x],x·stochvector(n-1))
EndIf
EndFunc
```

Anmerkung: Es hätte auch erste Komponente x und Fakor in der zweiten (1-x) sein können.

stochvector(1) ▸ [1]

stochvector(2) ▸ [0.056403 0.943597]

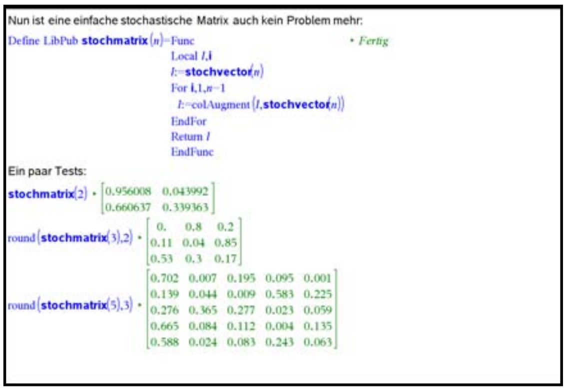
stochvector(3) ▸ [0.046942 0.813256 0.139802]

stochvector(4) ▸ [0.198595 0.290884 0.135894 0.374626]

Der Vektor wird in diesem Konzept rekursiv erzeugt und die Komponenten haben eine Verteilung mit dem mit dem Erwartungwert $\frac{1}{n}$.

Veranschaulichung für n=3 in einem Dreieck, das dann gleichverteilte Punkte haben muss, siehe unten.

1.4



Nun ist eine einfache stochastische Matrix auch kein Problem mehr:

Define LibPub stochmatrix(n)=Func ▸ Fertig

```
Local l,i
l:=stochvector(n)
For i,1,n-1
 l:=colAugment(l,stochvector(n))
EndFor
Return l
EndFunc
```

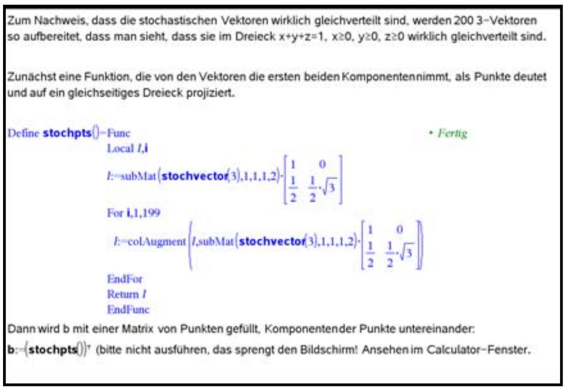
Ein paar Tests:

stochmatrix(2) ▸ $\begin{bmatrix}0.956008 & 0.043992\\0.660637 & 0.339363\end{bmatrix}$

round(stochmatrix(3),2) ▸ $\begin{bmatrix}0. & 0.8 & 0.2\\0.11 & 0.04 & 0.85\\0.53 & 0.3 & 0.17\end{bmatrix}$

round(stochmatrix(5),3) ▸ $\begin{bmatrix}0.702 & 0.007 & 0.195 & 0.095 & 0.001\\0.139 & 0.044 & 0.009 & 0.583 & 0.225\\0.276 & 0.365 & 0.277 & 0.023 & 0.059\\0.665 & 0.084 & 0.112 & 0.004 & 0.135\\0.588 & 0.024 & 0.083 & 0.243 & 0.063\end{bmatrix}$

1.5



Zum Nachweis, dass die stochastischen Vektoren wirklich gleichverteilt sind, werden 200 3-Vektoren so aufbereitet, dass man sieht, dass sie im Dreieck x+y+z=1, x≥0, y≥0, z≥0 wirklich gleichverteilt sind.

Zunächst eine Funktion, die von den Vektoren die ersten beiden Komponenten nimmt, als Punkte deutet und auf ein gleichseitiges Dreieck projiziert.

Define stochpts()=Func ▸ Fertig

Local l,i

l:=subMat(stochvector(3),1,1,1,2)·$\begin{bmatrix}1 & 0\\\frac{1}{2} & \frac{1}{2}\cdot\sqrt{3}\end{bmatrix}$

For i,1,199

 l:=colAugment$\left(l,\text{subMat}(\text{stochvector}(3),1,1,1,2)\cdot\begin{bmatrix}1 & 0\\\frac{1}{2} & \frac{1}{2}\cdot\sqrt{3}\end{bmatrix}\right)$

EndFor

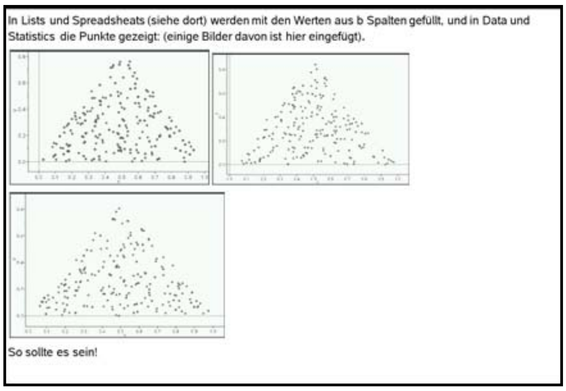
Return l

EndFunc

Dann wird b mit einer Matrix von Punkten gefüllt, Komponenten der Punkte untereinander:

b:=(stochpts())^T (bitte nicht ausführen, das sprengt den Bildschirm! Ansehen im Calculator-Fenster.

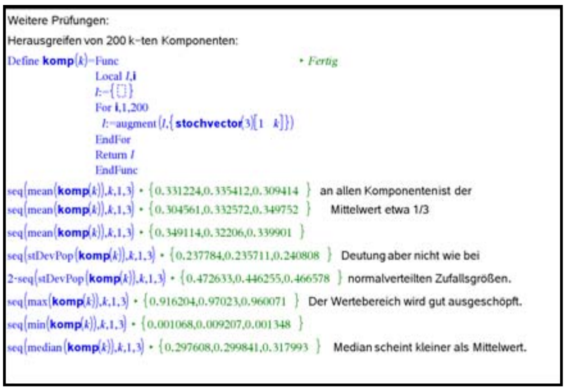
1.6



In Lists und Spreadsheats (siehe dort) werden mit den Werten aus b Spalten gefüllt, und in Data und Statistics die Punkte gezeigt: (einige Bilder davon ist hier eingefügt).

So sollte es sein!

1.7



Weitere Prüfungen:

Herausgreifen von 200 k-ten Komponenten:

Define komp(k)=Func ▸ Fertig

```
Local l,i
l:={□}
For i,1,200
 l:=augment(l,{stochvector(3)[1 k]})
EndFor
Return l
EndFunc
```

seq(mean(komp(k)),k,1,3) ▸ {0.331224,0.335412,0.309414} an allen Komponenten ist der

seq(mean(komp(k)),k,1,3) ▸ {0.304561,0.332572,0.349752} Mittelwert etwa 1/3

seq(mean(komp(k)),k,1,3) ▸ {0.349114,0.32206,0.339901}

seq(stDevPop(komp(k)),k,1,3) ▸ {0.237784,0.235711,0.240808} Deutung aber nicht wie bei

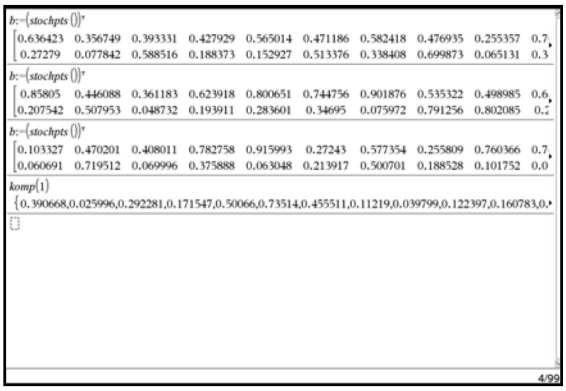
2·seq(stDevPop(komp(k)),k,1,3) ▸ {0.472633,0.446255,0.466578} normalverteilten Zufallsgrößen.

seq(max(komp(k)),k,1,3) ▸ {0.916204,0.97023,0.960071} Der Wertebereich wird gut ausgeschöpft.

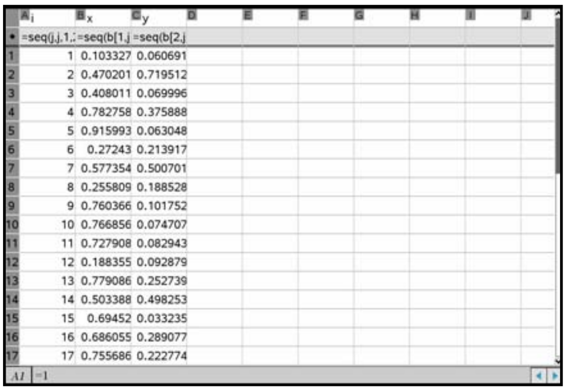
seq(min(komp(k)),k,1,3) ▸ {0.001068,0.009207,0.001348}

seq(median(komp(k)),k,1,3) ▸ {0.297608,0.299841,0.317993} Median scheint kleiner als Mittelwert.

1.8

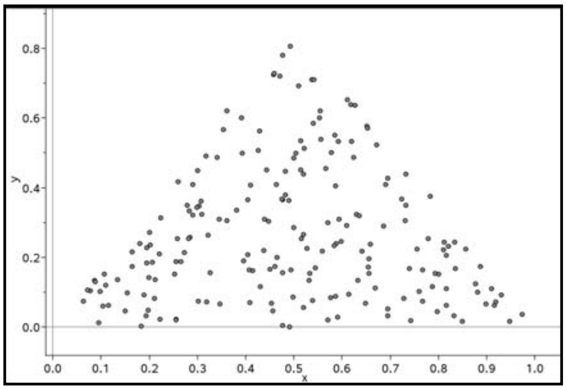


1.9

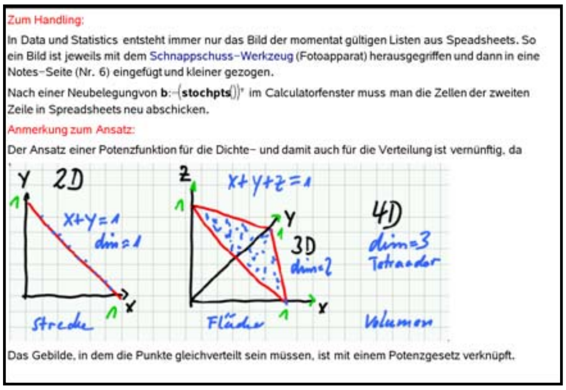


	A i	B x	C y
•	=seq(j,j,1,:	=seq(b[1,j	=seq(b[2,j
1	1	0.103327	0.060691
2	2	0.470201	0.719512
3	3	0.408011	0.069996
4	4	0.782758	0.375888
5	5	0.915993	0.063048
6	6	0.27243	0.213917
7	7	0.577354	0.500701
8	8	0.255809	0.188528
9	9	0.760366	0.101752
10	10	0.766856	0.074707
11	11	0.727908	0.082943
12	12	0.188355	0.092879
13	13	0.779086	0.252739
14	14	0.503388	0.498253
15	15	0.69452	0.033235
16	16	0.686055	0.289077
17	17	0.755686	0.222774

1.10



1.11



1.12