

tool

tool.tns ist eine Sammlung nützlicher Funktionen Haftendorn 2011

tool.tns muss in mylib stehen. Bibliotheken aktualisieren

stomat(k) erzeugt stochastische $n \times n$ Zufallsmatrix

Dabei sind die Zeilensummen 1. Gebrauch siehe Extraseite

liste := $\{h,a,n,s,e\} \triangleright \{h,a,n,s,e\}$

tier := **eldel**(**liste**,3) $\triangleright \{h,a,s,e\}$ **eldel**(k) löscht das k -te Element aus einer Liste,

liste bleibt unverändert! **liste** $\triangleright \{h,a,n,s,e\}$ **tier** $\triangleright \{h,a,s,e\}$

lidel(**liste**,2,3) $\triangleright \{h,s,e\}$ **lidel**(**liste**, k , r) löscht die Elemente k bis incl. r

perm(**liste**) $\triangleright \{s,e,a,h,n\}$ **perm**(**liste**) $\triangleright \{n,e,s,a,h\}$

erzeugt eine zufällige Permutation der Liste

Beispiel für stochastische Matrix stomat(n) ..

$$\mathbf{aa} := \mathbf{stomat}(3) \triangleright \begin{bmatrix} 0.64 & 0.34 & 0.02 \\ 0.88 & 0.12 & 0. \\ 0.28 & 0.28 & 0.44 \end{bmatrix} \quad \text{erzeugt stochastische } n \times n \text{ Zufallsmatrix}$$

Dabei sind die Zeilensummen 1 $\mathbf{start} := [10 \ 30 \ 40] \triangleright [10 \ 30 \ 40]$

$$\mathbf{start} \cdot \mathbf{aa} \triangleright [44. \ 18.2 \ 17.8] \quad \mathbf{aa}^{20} \triangleright \begin{bmatrix} 0.697622 & 0.277463 & 0.024915 \\ 0.697622 & 0.277463 & 0.024915 \\ 0.697622 & 0.277463 & 0.024915 \end{bmatrix}$$

Die $\mathbf{end} := \mathbf{start} \cdot \mathbf{aa}^{20} \triangleright [55.8097 \ 22.1971 \ 1.99321]$

$$\sum_{i=1}^3 (\mathbf{start}[1,i]) \triangleright 80 \quad \sum_{i=1}^3 (\mathbf{end}[1,i]) \triangleright 80. \quad \mathbf{aa}^{20} \text{ gibt die langfristige Verteilung der 80}$$

Leute an.

stomat	1/20
--------	------

```

Define LibPub stomat (n)=
Func
© stomat (n) gibt quadr. stochastische Zufalls-Matrix dim nxn aus.
Local i,j,ma,lima,li,z
ma:=newMat(n,n)
lima:={ }
For i,1,n © n Zeilen li werden erzeugt
z:=randInt(0,100) © volle Prozente, Division später
li:={ z }
For j,2,n-1
While sum(li)+z>100 © Wahrsch. nicht größer als 100%
z:=randInt(0,100)
EndWhile
li:=augment(li,{ z })
EndFor
z:=100-sum(li)
li:=augment(li,{ z }) © Hier ist eine Zeile fertig
tool\perm(li) © vermeidet, dass die großen Werte bevorzugt vorn sind
lima:=augment(lima,li) ©Lange Liste aller Werte
EndFor

```

```
eldel 2/2  
Define LibPub eldel (liste,k)=  
Func  
©eldel (liste,k) gibt Liste ohne das Element k zurück  
  Return augment (left(liste,k-1),right(liste,dim(liste)-k))  
EndFunc
```

```
lidel 2/2
Define LibPub lidel(liste,k,r)=
Func
©lidel(liste,k,r) gibt Liste ohne die Elemente k bis r zurück
  Return augment(left(liste,k-1),right(liste,dim(liste)-r))
EndFunc
```

```
perm 6/11
Define LibPub perm(liste)=
Func
© perm(liste) permutiert die Liste zufällig
Local i,e,z,n,pli
  n:=dim(liste)
  pli:={ }
  For i,1,n-1
    z:=randInt(1,dim(liste))
    e:=mid(liste,z,1)
    pli:=augment(pli,e)
    liste:=eldel(liste,z)
  EndFor
  Return augment(pli,liste)
EndFunc
```