

Mit dem Programm `Get_Sample.PRO` können unter IDL [1] zufällige Stichproben von Datensätzen (z. B. von EEG-Potentialen), die bestimmten Kriterien genügen, gezogen werden.

Bei jedem Aufruf (Zug) dieses Funktions-Unterprogramms wird eine vollständige Stichprobe (sample) „n aus m“ mit Hilfe der Funktion `Lotto` [3] ermittelt, wobei die aufeinanderfolgenden Züge (moves) zufällig sind. Die Stichprobe wird dabei durch ein zweidimensionales Feld mit Indexwerten dargestellt.

Das Programm steht im Internet zur Verfügung. Diese Beschreibung gibt den Stand der Version V 01.4 (= V 1.0 – Edit 4) wieder. Es wurde von der Abt. Wiss. Datenverarbeitung in der „Interactive Data Language“ IDL [1, 2] angefertigt und kann ggf. vom jeweiligen Anwender selbst modifiziert werden [Projekt: C92 250].

Das Programm steht im Internet zur Verfügung. Diese Beschreibung gibt den Stand der Version V 01.4 (= V 1.0 – Edit 4) wieder. Es wurde von der Abt. Wiss. Datenverarbeitung in der „Interactive Data Language“ IDL [1, 2] angefertigt und kann ggf. vom jeweiligen Anwender selbst modifiziert werden [Projekt: C92 250].

1. Beschreibung:

Das Programm `Get_Sample` ermittelt die zufallsausgewählten Stichproben mit folgendem Algorithmus:

```
p = m/n ; z. B.: 64/16 = 4
q = n*n/m ; z. B.: 16*16/64 = 4

Repeat Begin
  index = Lotto( n, m, start, /CUT)
  h = Histogram( index(*,1), Min=0, Max=p-1)
  t = Where( h eq q, count) ; Gleichverteilung?
EndRep Until (count eq p) ; Der 2. Index ist nun
                          ; gleichverteilt.
Return, index ; Stichprobe übergeben.
```

In der Repeat-Schleife wird zunächst mit `Lotto` eine Stichprobe gezogen und die Indexwerte als zweidimensionales Feld `index` der Größe (n, 2) übergeben. Von den Indexwerten `index(*, 1)` wird dann die Dichtefunktion `h` (Histogramm) mit den Klassen `[0..q-1]` berechnet und geprüft, ob alle Klassen gleich besetzt sind. Ist dieses nicht der Fall (`count # p`), dann wird die gezogene Stichprobe verworfen und eine neue Stichprobe gezogen (siehe auch Hinweis unter Punkt 4).

Sobald die gewünschte Gleichverteilung des zweiten Parameters vorliegt (`count = p`), wird die Repeat-Schleife verlassen und die Stichprobe im zweidimensionalen Feld `index` an das rufende Programm übergeben, wobei `index(*, 0)` n Indexwerte `[0..n-1]` und `index(*, 1)` n Indexwerte `[0..q-1]` enthält.

2. Aufruf:

`Get_Sample (n, m)`

2.1 Variablen (VAR):

Die Variablen im Aufruf von `Get_Sample` haben folgende Bedeutung:

n — Umfang der Stichprobe (sample).

m — Gesamtanzahl der möglichen Stichproben. Dieses **m** muß als Produkt aus **n** und der Anzahl der Möglichkeiten des zweiten Stichproben-Parameters gewählt werden.

Beispiel: Bei den EEG-Potentialen ist der zweite Stichproben-Parameter die Reizart (categorie). Davon existieren z. B. 4 Möglichkeiten; dann ist $m = n \times 4$ zu wählen.

2.2 Keywords:

Durch die Angabe folgender Keywords kann der Ablauf des Programms modifiziert werden.

`/EQUAL_2` — Das ist zur Zeit die einzige Einstellung. Deshalb ist die Angabe dieses Keywords nicht erforderlich.

Es werden intern so lange Stichproben gezogen, bis die zweiten Indizes gleichverteilt sind. Diese sind die Zahlen der zweiten Komponente `z(*, 1)`, die von der Funktion `Lotto` bei Angabe des Keywords `/CUT` erzeugt werden [3]. Sie liegen im Bereich `[0..q-1]`, wobei $q = (m \text{ DIV } n)$ ist.

Ein Beispiel: Beim „16 aus 64“ ist $q = 4$ und der Bereich des zweiten Index ist `[0..3]`. Der Bereich des ersten Index ist dabei `[0..15]`, da $n = 16$ ist.

2.3 Beispiel(e):

Im IDL-Programm `Ave_EEG` [4] wird das Programm `Get_Sample` unter dem Menü-Punkt »Perform random shuffling« zur Ermittlung von zufallsausgewählten Stichproben von EEG-Potentialen eingesetzt. Im folgenden ist ein Auszug dieser Anwendung wiedergegeben, der die Verwendung von `Get_Sample` deutlich macht.

```
; Ziehen einer Stichprobe:
i_samp = Get_Sample(nave, Fix(ncat*nave))
...
; Basis-Ordnungsnummern berechnen:
order = n_loc*(ncat*i_samp(*,0) + i_samp(*,1))
...
; Ein Potential extrahieren:
sp1 = sp((order(j-1) + num0 - 1)*nx: -
         ((order(j-1) + num0)*nx - 1))
```

Die jeweils gezogene Stichprobe wird in dem zweidimensionalen Feld `i_samp` der Größe (nave, 2) übergeben, wobei hier in `i_samp(*, 0)` die Indexwerte `i_vpn` für die Versuchspersonen und in `i_samp(*, 1)` die Indexwerte `i_cat` für die Reizarten (categories) gespeichert sind.

Beim Ziehen der Stichprobe stellt hierbei `Get_Sample` sicher, daß innerhalb jeder gezogenen Stichprobe die auftretenden Reizarten (categories) immer gleichverteilt sind.

Aus den Indexwerten des Feldes `i_samp` werden dann mit Hilfe der geltenden Speicherabbildungsfunktion die Basis-Ordnungsnummern `order` der EEG-Potentiale ermittelt, die dann in den weiteren Berechnungen (z. B. beim Extrahieren eines bestimmten Potentials) verwendet werden.

3. Erforderliche Programme:

Dieses Programm verwendet folgende IDL-Programme der Abt. Wiss. Datenverarbeitung:

- **Lotto** (P 476/35): Erzeugen einer Zufallszahlenfolge „n aus m“ nach dem Lotto-Prinzip. Es wird ein Satz von n Zahlen aus dem Bereich `[1..m]` gezogen [3].

4. Sonstige Hinweise:

- Ab V 01.5 wurde für den 2. Index eine Quasi-Gleichverteilung durch die Anweisung `t = Where((h eq q) or (h eq q+1), count)` zugelassen.

5. Literatur:

- [1] Research Systems, Inc.: IDL – Interactive Data Language. Version 2.1. Boulder (USA): RSI 1991 (Edition vom 2.4.1991). E-Mail: idl@boulder.colorado.edu.
- [2] Dittberner, K.-H.: Interactive Data Language IDL: Eine Einführung. FU Berlin (IfP): wdv-notes Nr. 85, 1988–1992.
- [3] Dittberner, K.-H.: IDL-Programm: Lotto.PRO (P 476/35). FU Berlin (IfP): wdv-notes Nr. 107, 1992–1993.
- [4] Dittberner, K.-H.: IDL-Programm: Ave_EEG.PRO (P 476/37). FU Berlin (IfP): wdv-notes Nr. 102, 1992–1993.

6. Copyright-Notiz:

© 1992-1994 – K.-H. Dittberner c/o Freie Universität Berlin. All rights reserved. No part of this software package may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any form by any means without the written permission of Karl-Heinz Dittberner. Permission will be granted for non-profit redistribution.