

Mit dem Programm Lotto.PRO können unter IDL [1] unabhängige Zufallszahlenfolgen „ n aus m “ nach dem Lotto-Prinzip erzeugt werden.

Bei jedem Aufruf (Zug) dieses Funktions-Unterprogramms wird ein Satz (set) von n ganzen Zahlen aus dem Bereich

[1.. m] gezogen, wobei die aufeinanderfolgenden Züge (moves) zufällig sind. Die gezogenen Zahlen entstammen einer gleichverteilten Grundgesamtheit.

Das Programm steht im Internet zur Verfügung. Diese Beschreibung gibt den Stand des Programms in der Version V 01.5

(= V 1.0 – Edit 5) 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 bzw. ergänzt werden [Projekt: C92 250].

1. Beschreibung:

Dieses Programm ist keine „Spielerei“! Es wird zum Beispiel zum Ermitteln zufallsausgewählter Stichproben von wissenschaftlichen Daten benötigt [4].

Das Programm *Lotto* verwendet zur Ermittlung der Zufallszahlen [3] die IDL-Funktion $z = \text{RandomU}(start)$. Diese Funktion erzeugt nahezu gleichverteilte Zufallszahlen vom Real-Typ im Bereich ($0 \leq z < 1.0$). Die so erzeugten Zufallszahlen werden dann durch lineare Transformation in Integer-Zahlen z , die im Bereich ($1 \leq z \leq m$) liegen, umgewandelt.

Bei einem Aufruf von *Lotto* werden insgesamt n solcher Zahlen gezogen. Dabei ist sichergestellt, daß in einem Zug – wie beim Zahlenlotto – keine Doubletten auftreten können. Das Programm *Lotto* bietet außerdem die Option, die gezogenen Zahlen in zwei Komponenten zu zerlegen (siehe Punkt 2.2).

2. Aufruf:

Lotto (n , m [, $start$])

2.1 Variablen (VAR):

n — Anzahl der Zahlen in dem Zufallszahlenset „ n aus m “.

m — Obere Grenze des Zufallszahlenbereichs [1.. m]. Diese sollte ein ganzzahlig Vielfaches von n sein.

start — Mit dieser Variablen wird der Startpunkt in der Zufallszahlenfolge bestimmt. Ist diese im Aufruf angegeben und bleibt diese im übergeordneten Programm unverändert, wird bei jedem Aufruf von *Lotto* die beim ersten Aufruf begonnene Zufallszahlenfolge fortgesetzt und keine neue Folge begonnen. Wird im Aufruf die Variable *start* nicht angegeben, dann wird der Startpunkt aus der aktuellen System-Uhrzeit abgeleitet.

2.2 Keywords:

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

/Cut — Jede erzeugte Zufallszahl wird in zwei Komponenten zerlegt. Die Zahlen der ersten Komponente $z(*,0)$ liegen im Bereich [0.. $n-1$]. Die Zahlen der zweiten Komponente $z(*,1)$ liegen im Bereich [0.. $q-1$] mit $q = (m \text{ DIV } n)$. Beispiel: Beim „16 aus 64“ ist [0..15] der Bereich der ersten Komponente und [0..3] der Bereich der zweiten Komponente. Diese Komponenten können zur bequemen Indizierung von Variablen in einem übergeordneten Programm verwendet werden.

/Same — Durch Angabe dieses Keywords kann für besondere Zwecke (z. B. Tests) eine immer identische Folge der Zufallszahlen erzeugt werden. Eine Angabe der Variablen *start* wird dabei ignoriert.

2.3 Beispiel(e):

Der dem Programm *Lotto* zugrundeliegende Algorithmus wurde mit einem speziellen Testprogramm [6] unter *Mathematica* [5] untersucht. Dabei wurden u. a. n Zufallszahlensets, jeweils bestehend aus 16 Zahlen aus dem Bereich [1.. m], in fortlaufender Folge gezogen. Die Häufigkeit des Auftretens aller gezogenen Einzelwerte ist im Histogramm der Abb. 1 dargestellt.

Bei $n = 2000$ solcher aus dem Bereich [1..64] gezogener Zufallszahlensets zu je 16 Zufallszahlen ist der Erwartungswert des mittleren Auftretens einer jeden möglichen Zahl gleich $16 \times n/m$, d. h. $16 \times 2000/64 = 500$. Der Test ergab, daß bei $n = 2000$ Zügen alle Zahlen aus dem Bereich [1..64] – wie erwartet – im Mittel genau 500 mal auftreten.

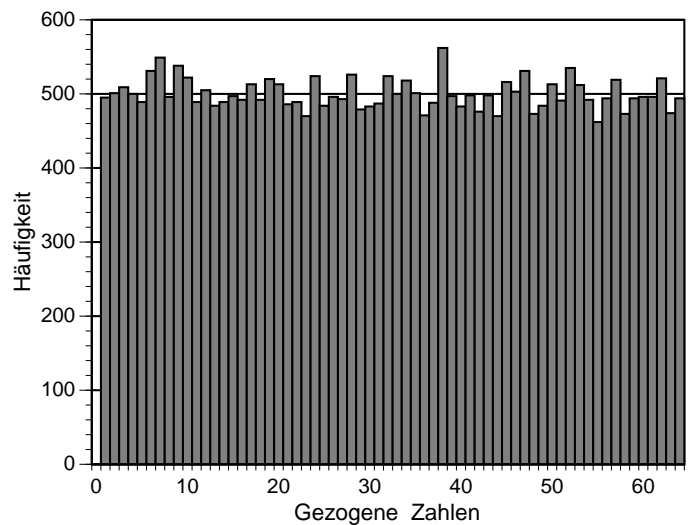


Abb. 1: Histogramm von 2000 Ziehungen „16 aus 64“ [6].
Der Mittelwert aller Häufigkeiten ist 500.
Die Standardabweichung ist $\pm 20,2$.

3. Erforderliche Programme:

Dieses Programm verwendet die folgenden IDL-Funktionen: Fix, RandomU, Where.

4. Sonstige Hinweise:

■ Die Abb. 1 wurde probeweise mit der Grafik-Applikation »DeltaGraph« V 1.5c auf einem MAC Quadra-700 angefertigt. Und »DeltaGraph« überzeugt dabei nicht.

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, 1988–1993.
- [3] Park and Miller: Random Number Generators: Good ones are hard to find. Comm. ACM, Vol. 31, No. 10, 1192 (Oct. 1988).
- [4] Dittberner, K.-H.: IDL-Programm: Get_Sample.PRO (P 476/39). FU Berlin (IfP): wdv-notes Nr. 108, 1992–1994.
- [5] Dittberner, K.-H.: MAC-Software: Mathematica V 2.0. FU Berlin (IfP): wdv-notes Nr. 71, 1990–1992.
- [6] Dittberner, K.-H.: Tests für ein »Random Shuffling«. Mathematica-Notebook vom 23. April 1992. Projekt: C92 250.

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.