

Übung zur Vorlesung
Computergestützte Statistik
Wintersemester 2022/2023
Übungsblatt Nr. 7

Abgabe ist Dienstag, der 29.11.2022 bis 08:00 Uhr im Moodle

Hinweis: Dies ist das letzte Blatt der ersten Hälfte. Wir werden Ihnen mit der Korrektur dieses Blatts mitteilen, ob sie die zur Zulassung benötigten 50% der Punkte auf Blatt 1-7 erreicht haben. Bitte denken Sie daran, dass sie auch auf Blatt 8-14 nochmals 50% der Punkte benötigen.

Aufgabe 1 (4 Punkte)

Erzeugen Sie diskret gleichverteilte Zufallszahlen. Gehen Sie wie folgt vor:

- a) (2 Punkte) Implementieren Sie den erweiterten euklidischen Algorithmus (Beispiel 2.2) zur Bestimmung eines inversen Elements.
- b) (2 Punkte) Verwenden Sie a) um den Inversionsgenerator (Definition 2.9) zu implementieren.

Testen Sie Ihre Funktion, indem Sie einige Zufallszahlen erzeugen und ihre Verteilung visualisieren. Achten Sie wie üblich auf eine ordentliche Dokumentation Ihrer Funktionen.

Aufgabe 2 (4 Punkte)

Überprüfen Sie, ob der Inversionsgenerator aus Aufgabe 1 mit den Einstellungen aus Tabelle 1 gute, gleichverteilte Zufallszahlen erzeugt. Falls Sie Aufgabe 1 nicht bearbeitet haben, nutzen sie bitte stattdessen einmal `runif(1e6)` und einmal den LCF Generator aus der Datei `LCF.R` mit $a = 12353243123$, $c = 453816693$ und $m = 2^{31}$.

- a) (1 Punkt) Implementieren Sie den Permutationstest,
- b) (1 Punkt) Implementieren Sie den Maximum- τ -Test. (Einen implementierten Gleichverteilungstest finden Sie ebenfalls in `LCF.R`.)
- c) (2 Punkte) Finden Sie heraus, ob ihr Generator aus Aufgabe 1 gute Ergebnisse liefert. Überprüfen Sie außerdem, ob die Zufallszahlen in der Datei `zufall.RData` gute, gleichverteilte Zufallszahlen sind. Nutzen Sie dazu Ihre Tests aus a) und b). Gehen Sie wie folgt vor:
 - i) Erzeugen Sie `1e06` Zufallszahlen mit Ihrem Generator für beide Einstellungen. Um gleichverteilte Zufallszahlen auf $[0, 1]$ zu erhalten, teilen Sie diese jeweils durch m .
 - ii) Führen Sie auf 1000 Blöcken mit je 1000 Zufallszahlen die Tests durch. Verwenden Sie wie im Skript vorgeschlagen $T = 3, 4, 5$, $\tau = 2, 3, 4, 5$ und $D = 32$.
 - iii) Wiederholen Sie Teil ii) für die beiden Vektoren von Zufallszahlen aus der Datei `zufall.RData`.
 - iv) Stellen Sie die Ergebnisse aus ii) und iii) graphisch dar und beantworten sie damit, ob es sich jeweils um gute Zufallszahlen handelt bzw. ggf. warum die Zufallszahlen nicht gut sind.

Hinweis: Vielleicht hilft Ihnen die Funktion `permutations` aus dem Paket `e1071` weiter. Sie dürfen außerdem die R-Funktion `chisq.test` verwenden und müssen keinen χ^2 -Test selbst implementieren. Die Warnungen „Chi-squared approximation may be incorrect“ dürfen Sie ignorieren.

a	c	m
14288	758634	$2^{31} - 1$
10	10	$2^{31} - 1$

Tabelle 1: Einstellungen für den Inversionsgenerator

Aufgabe 3

(4 Punkte)

Verwenden Sie `runif(n, 0, 1)`, um `n` gleichverteilte Zufallszahlen auf $[0, 1]$ zu erzeugen und somit einen der einfachen Zufallsgeneratoren zu simulieren. Implementieren Sie Generatoren für die folgenden Verteilungen (jeweils 1 Punkt):

- a) Poissonverteilung,
- b) Exponentialverteilung,
- c) Binomialverteilung,
- d) Dreiecksverteilung.

Veranschaulichen Sie für jeden Generator, dass er Zufallszahlen aus der jeweiligen Verteilung erzeugt. Sie dürfen `dpois`, `dexp`, etc. nutzen, aber natürlich **nicht** `rpois`, `rexp`, etc.