# Blatt 1

Prof. Dr. Reinhard Höpfner Dipl.-Math. Michael Diether

**Aufgabe 0**

Frischen Sie Ihre Kenntnisse in `R` auf, indem Sie etwa auf `www.r-project.org` die „Introduction to `R`“, Abschnitte 2 („Simple Manipulations, Numbers and Vectors“), 8 („Probability Distributions“) und 10 („Writing your own Functions“), lesen. Auch ein Blick in Abschnitt 12 („Graphical Procedures“) kann nützlich sein.

**Aufgabe 1**

*a*) Generieren Sie zwei Datensätze `A` und `B` mit jeweils 10.000 $\Gamma(2,1)$-verteilten Pseudozufallszahlen $X_1, \ldots, X_{10000}$ und plotten Sie die Paare $(i, X_i)$, $i = 1, \ldots, 10.000$, wobei die $X_i$ die Werte aus den Datensätzen `A` bzw. `B` seien.

*b*) Vergleichen Sie jeweils für $n = 250, 1.000, 10.000$ ein Histogramm der ersten $n$ Beobachtungen mit der Dichte von $\Gamma(2,1)$. Passen Sie die Zellbreite der Histogramme geeignet an.

**Aufgabe 2 (Illustration des starkes Gesetzes der großen Zahlen)**

Generieren Sie zwei Datensätze `A` und `B` von jeweils 10.000 $\mathcal{P}(3)$-verteilten Pseudozufallszahlen $X_1, \ldots, X_{10000}$. Plotten Sie die Pfade $n \mapsto \frac{1}{n}\sum_{i=1}^{n} X_i$. Vergleichen Sie mit den Pfaden, die nur die ersten 50 bzw. 250 Zufallszahlen berücksichtigen.

**Aufgabe 3**

Was ändert sich, wenn man in den Aufgaben 1 und 2 die Cauchyverteilung verwendet? Woran liegt das?

**Aufgabe 4 (Zum Satz von Glivenko-Cantelli)**

*a*) Generieren Sie einen Datensatz von 10.000 $\mathcal{N}(0,1)$-verteilten Pseudozufallszahlen. Vergleichen Sie jeweils für die ersten $n = 25, 100, 250, 1.000, 5.000, 10.000$ Elemente der Datensätze die empirischen Verteilungsfunktionen mit der Verteilungsfunktion $\Phi$ der Standardnormalverteilung. Verwenden Sie hierzu den Befehl `ecdf`, der von `R` bereit gestellt wird.

*b*) Schreiben Sie eine Funktion, die „zu Fuß“ (d. h. ohne die Funktion `ecdf` zu verwenden) die empirische Verteilungsfunktion eines Datensatzes berechnet und ein mathematisch möglichst präzises Bild ausgibt (also eine rechtsstetige stückweise konstante Funktion). Vergleichen Sie mit Hilfe des Befehls `system.time` die Geschwindigkeit Ihrer Funktion mit der Performance aus Teil *a*).

*Hinweis*: Die empirische Verteilungsfunktion lässt sich leichter berechnen, wenn die Beobachtungen aufsteigend der Größe nach sortiert werden. Die Sprungstellen können Sie dann der Reihe nach aus einem Vektor auslesen.

**Bitte wenden!**

**Aufgabe 5**

*a*) Visualisieren Sie durch einen `persp`-Plot einige log-Likelihood-Flächen basierend auf den i.i.d. Beobachtungen $X_1, \dots, X_n$ einer $\mathcal{N}(\mu_0, \sigma_0^2)$-Verteilung

- mit einer kleinen Zahl von Beobachtungen, etwa $n = 5$,
- mit ein wenig mehr Beobachtungen, etwa $n = 20$,
- mit einer größeren Zahl von Beobachtungen, etwa $n = 100$.

Wählen Sie für Ihre Simulation $\mu_0 = \frac{1}{2}$ und $\sigma_0^2 = \frac{1}{2}$.

*b*) Visualisieren Sie nun die log-Likelihood-Flächen zu den gleichen Beobachtungen $X_1, \dots, X_n$ wie in Teil *a*) mit einem `contour`-Plot, der um den Maximum-Likelihood-Schätzer

$$(\hat{\mu}, \hat{\sigma}_0^2) = \left( \frac{1}{n} \sum_{i=1}^{n} X_i, \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})^2 \right)$$

zentriert ist. Geben Sie diesen Maximum-Likelihood-Schätzer sowie den „wahren“ Parameter $(\mu_0, \sigma_0^2)$ in Ihrem Plot mit aus.

*Hinweise*:

1. Konsultieren Sie die `R`-Hilfe für die Feineinstellung der Graphik-Parameter.
2. Arbeiten Sie für den `persp`-Plot mit dem Parameter `zlim`, und für den `contour`-Plot mit dem Parameter `levels`, damit der interessante Bereich der Plots (in der Nähe des Maximums der log-Likelihood-Fläche) sichtbar wird.
3. Achten Sie darauf, dass die Rasterung der Graphik nicht zu grob ist, um „eckige“ Plots zu vermeiden. Da ein feines Raster die Laufzeit des Programms stark beeinträchtigen kann, sollten Sie zu Testzwecken zunächst auf groben Rastern rechnen.

---

**Abgabe:** Per E-Mail an `diethermm@uni-mainz.de` bis Montag, 2. 5. 2011, 23.59 Uhr