

3. Übungsblatt zur Vorlesung Lineare Optimierung

1. Aufgabe (loops und vectorized calculation in R): Gegeben sei die Summe

$$s_n := \sum_{k=1}^n (2k-1) = 1 + 3 + 5 + 7 + \dots + 2n - 1 \quad (1)$$

- Berechnen Sie die Summen s_1, s_2, s_3, s_4 und s_5 mit Bleistift und Papier und finden Sie eine allgemeine Formel für s_n . Beweisen Sie diese Formel mit vollständiger Induktion.
- Legen Sie die Variable $n = 100$ in R an und erzeugen Sie den Vektor $\vec{x} := (1, 3, 5, \dots, 2n - 1) \in \mathbb{R}^n$.
- Machen Sie sich mit dem R-Befehl `cumulative sum`, `cumsum()`, vertraut. Dazu können Sie die R-Hilfe bemühen mit der Syntax `?cumsum()`. Erzeugen Sie dann den Vektor

$$\vec{s} := (s_1, s_2, s_3, \dots, s_n) \quad (2)$$

wobei die s_n 's durch die Formel (1) gegeben sind.

- Ein R-loop hat die Syntax

```
for(k in vector)
{
  ...do something...
}
```

und die Variable k nimmt dann nacheinander die Werte `vector[1]`, `vector[2]`, `vector[3]`, ... an. Das müssen nicht immer numerische Werte sein, sondern können etwa auch Text-Variablen sein. Erzeugen Sie dann noch einmal den Vektor \vec{s} aus (2) mit Hilfe eines loops, also ohne die Funktion `cumsum()` zu benutzen.

- Setzen Sie jetzt $n = 4'000'000$ und wiederholen Sie die Rechnungen aus Teil (c) und (d). Bestimmen Sie jeweils die Rechenzeiten. Das können Sie etwa mit Hilfe der `Sys.time()`-Funktion machen, wie genau?

2. Aufgabe: Legen Sie die Variablen $n = 50$ und $m = 500$ in R an und erzeugen Sie dann eine Matrix $A \in \mathbb{R}^{m \times n}$, für die jedes Element $a_{i,j}$ durch eine auf dem Intervall $[1, 2]$ gleichverteilte Zufallszahl gegeben ist. Erinnern Sie sich daran, was der Rang einer Matrix `rank(A)` ist. Versuchen Sie dann eine R-Funktion zu finden, die den Rang einer Matrix berechnet. Dazu ist es sinnvoll, die Hilfe-Seiten mit der Syntax `??Thema` aufzurufen, da die Syntax `?Thema` nur nach der genauen Funktion `Thema()` sucht, Sie müssen den Funktionsnamen da also schon kennen.

..*bitte wenden*

- a) Berechnen Sie den Rang Ihrer Matrix A .
- b) Berechnen Sie den Rang für 1000 verschiedene Realisierungen von A und speichern Sie diese Zahlen in einem Vektor `rangA` der Länge 1000. Gibt es eine Realisierung, für die der Rang ungleich 50 ist? Für das Testen von Vektorelementen auf einen gegebenen Wert könnte etwa die `ifelse()`-Funktion hilfreich sein.