

Übungsblatt 1

Einführung in Python

Übung 1.1 Schreiben Sie ein Python-Script, das folgende Funktionen enthält:

- eine Funktion *median(a)*, die zur übergebenen Liste *a* von Zahlen den Median zurückgibt
- eine Funktion *mean_stddev(a)* die zur übergebenen Liste *a* von Zahlen den Mittelwert und Standardabweichung zurückgibt (die Wurzelfunktion ist *math.sqrt*)

Rufen Sie diese Funktionen im Script nach der Funktionsdefinition an einer Testfolge auf und geben Sie das Ergebnis mit *print* aus.

Übung 1.2 Ein verbreitetes Programm zur grafischen Darstellung von Datensätzen ist *gnuplot*. Datensätze können interaktiv mit entsprechenden Befehlen geladen und dargestellt werden, was etwas umständlich ist. Bequemer ist es, Datensätze und Befehle zusammen in eine Datei zu schreiben, die z.B. so aussehen kann:

```
plot '-' with lines title 'sin'
0.1 0.0998
0.2 0.1987
0.3 0.2955
e
```

Wenn die Datei z.B. *plot.dat* heißt, kann Sie mit dem Befehl *gnuplot -persist plot.dat* dargestellt werden. Wenn man solch Datei automatisch mit einem Python-Script generiert, kann man sie auch nach stdout schreiben (z.B. mit *print*) und direkt nach *gnuplot* pipen:

```
python script.py | gnuplot -persist
```

Stellen Sie die Sinus-Funktion (*math.sin*) mit *gnuplot* dar, indem Sie die Werte über ein Python-Script generieren für $x = 0.0, 0.01, \dots, 3.99, 4.0$.

Übung 1.3 Um eine Grafik zu Drucken oder sie in andere Dokumente einzubinden, muss man mittels *gnuplot* eine EPS (encapsulated Postscript) Datei erzeugen. Die folgenden Befehle zu Beginn der *gnuplot*-Steuerdatei erzeugen statt einer Bildschirmanzeige die EPS-Datei *plot.eps*:

```
set term postscript eps
set output 'plot.eps'
```

Eine EPS-Datei kann mit *lpr* gedruckt und mit *gv* angesehen werden.

Erweitern Sie Ihr Script der vorherigen Aufgabe um eine optionale Kommandozeilenoption "-eps", die den obigen Zusatz in der Ausgabe bewirkt und schauen Sie sich die erzeugte EPS-Datei mit *gv plot.eps* an.