Aufgabe 1 Sehr viele Anwendungen wie z.B. Monte-Carlo-Simulationen benötigen stetige gleichverteilte Zufallszahlen. Entsprechend gibt es verschiedene Algorithmen um gleichverteilte Pseudozufallsvariablen zu generieren. Der Lehmer Algorithmus z.B., generiert rekursive Zufallszahlen mit der Formel

$$X_{k+1} = a \cdot X_k \mod m\,.$$

Im Jupyter-Notebook `Start_Hilfe.ipynb` auf Moodle finden Sie eine Implementierung; es werden mit dem Lehmer Algorithmus 100 Zahlen zwischen 0 und 1 generiert.

Überprüfen Sie rein visuell mit Hilfe von deskriptiver Statistik, ob die generierten Zahlen tatsächlich uniformverteilt sind oder nicht, (d.h. machen Sie einen geeigneten Histogram, einen Plot für ECDF und einen geeigneten Q-Q-Plot).

Machen Sie nun eine ähnliche Überprüfung für Zufallszahlen, die mit Numpy generiert werden. Mit Numpy können 100 gleichverteilte Zufallszahlen zwischen 0 und 1 so generiert werden

```
> np.random.uniform(0, 1, size = 100)
```

Lösungen: siehe Jupyter Notebook auf Moodle für die Lösungen

Aufgabe 2 Drücken Sie die folgenden Wahrscheinlichkeiten einer standardnormalverteilten Zufallsvariable Z mit Hilfe der Funktion Φ aus, und berechnen Sie die Wahrscheinlichkeiten mit Python und von Hand mit der Tabelle für Normalverteilung (auf Moodle zu finden).

(i) $P(Z \leq 1)$

(ii) $P(-1 \leq Z \leq 1)$

(iii) $P(|Z| \geq 0.5)$

(iv) $P(-3 \leq Z \leq 1)$

Lösungen

(i) Mit der Tabelle $P(Z \leq 1) = \Phi(1) = 8413$. In Python:

```
> stats.norm.cdf(1.0, loc = 0, scale = 1)
```

(ii) $P(-1 \leq Z \leq 1) = P(Z \leq 1) - P(Z \leq -1) = \Phi(1) - \Phi(-1) = 2\Phi(1) - 1 \approx 0.6827$ (vgl. Faustregel!) In Python:

```
> stats.norm.cdf(1.0) - stats.norm.cdf(-1.0)
```

(iii) Mit der Tabelle

$$\begin{aligned} P(|Z| \geq 0.5) &= 1 - P(-0.5 \leq Z \leq 0.5) \\ &= 2 - 2\Phi(0.5) = 2 - 2 \cdot 0.6915 \approx 0.6171 \end{aligned}$$

In Python:

```
> 1-(stats.norm.cdf(0.5) - stats.norm.cdf(-0.5))
```

(iv) Mit der Tabelle

$$\begin{aligned} P(-3 \leq Z \leq 1) &= \Phi(1) - \Phi(-3) = \Phi(1) + \Phi(3) - 1 \\ &= 0.8413 + 0.9987 - 1 \approx 0.84 \end{aligned}$$

In Python:

```
> stats.norm.cdf(1.0) - stats.norm.cdf(-3)
```

Berechnen Sie die folgenden Quantile in einem symmetrischen Interval um 0 herum.

(i) $P(x_1 \leq Z \leq x_2) \approx 0.6827$

(ii) $P(x_1 \leq Z \leq x_2) \approx 0.9545$

(iii) $P(x_1 \leq Z \leq x_2) \approx 0.9973$

Lösungen

(i) Hier ist approximativ der 1-sigma Bereich gefragt (vgl. Faustregel!), also $P(x_1 \leq Z \leq x_2) \approx 0.6827$ gilt für $x_1 = -1$ und $x_2 = 1$.

Mit der Tabelle

$$\begin{aligned} \Phi^{-1}(0.5 + 0.6827/2) &= \Phi^{-1}(0.84135) = 1 \\ \Phi^{-1}(0.5 - 0.6827/2) &= -1 \end{aligned}$$

In Python:

```
> stats.norm.ppf(0.5 - 0.6827/2, loc = 0, scale = 1)
> stats.norm.ppf(0.5 + 0.6827/2, loc = 0, scale = 1)
```

Achtung: es können Fehler bei der Rundung entstehen!

(ii) Hier ist approximativ der 2-sigma Bereich gefragt (vgl. Faustregel!), also $P(x_1 \leq Z \leq x_2) \approx 0.9545$ gilt für $x_1 = -2$ und $x_2 = 2$.

Mit der Tabelle

$$\begin{aligned} \Phi^{-1}(0.5 + 0.9545/2) &= \Phi^{-1}(0.97725) = 2 \\ \Phi^{-1}(0.5 - 0.9545/2) &= -2 \end{aligned}$$

In Python:

```
> stats.norm.ppf(0.5 - 0.9545/2, loc = 0, scale = 1)
> stats.norm.ppf(0.5 + 0.9545/2, loc = 0, scale = 1)
```

Achtung: es können Fehler bei der Rundung entstehen!

(iii) Hier ist approximativ der 3-sigma Bereich gefragt (vgl. Faustregel!), also $P(x_1 \leq Z \leq x_2) \approx 0.9973$ gilt für $x_1 = -3$ und $x_2 = 3$.

Mit der Tabelle

$$\begin{aligned} \Phi^{-1}(0.5 + 0.9973/2) &= \Phi^{-1}(0.99865) = 3 \\ \Phi^{-1}(0.5 - 0.9973/2) &= -3 \end{aligned}$$

In Python:

```
> stats.norm.ppf(0.5 - 0.9973/2, loc = 0, scale = 1)
> stats.norm.ppf(0.5 + 0.9973/2, loc = 0, scale = 1)
```

Achtung: es können Fehler bei der Rundung entstehen!

Aufgabe 3 Drücken Sie die folgenden Wahrscheinlichkeiten mit Hilfe der Funktion Φ aus, und berechnen Sie die Wahrscheinlichkeiten mit Python und mit der Tabelle für die Normalverteilung.

(i) $P(X \leq 2)$, wobei X normalverteilt mit $\mu = 1$ und $\sigma^2 = 9$ ist, d.h. $X \sim \mathcal{N}(\mu, \sigma^2)$.

(ii) $P(-4 \leq X \leq 8)$, wobei X normalverteilt mit $\mu = 2$ und $\sigma = 2$ ist. Das wird vielfach auch so geschrieben $X \sim \mathcal{N}(2, 4)$.

(iii) $P(|X| \leq 1)$, wobei X normalverteilt mit $\mu = -11$ und $\sigma^2 = 16$ ist. Das wird vielfach so geschrieben $X \sim \mathcal{N}(-11, 16)$.

Lösungen

(i) $P(X \leq 2) = P\left(Z \leq \frac{2-1}{3}\right) = \Phi\left(\frac{1}{3}\right) \approx 0.6306$

In Python:

```
> stats.norm.cdf(2, loc = 1, scale = 3)
```

(ii) $P(-4 \leq X \leq 8) = P\left(\frac{-4-2}{2} \leq Z \leq \frac{8-2}{2}\right) = 2\Phi(3) - 1 \approx 0.9973$

In Python:

```
> stats.norm.cdf(8, 2, 2) - stats.norm.cdf(-4, 2, 2)
```

(iii) $P(|X| \leq 1) = P\left(\frac{-1+11}{4} \leq Z \leq \frac{1+11}{4}\right) = \Phi(3) - \Phi(2.5) \approx 0.0049$

In Python:

```
> stats.norm.cdf(1, -11, 4) - stats.norm.cdf(-1, -11,  4)
```

Aufgabe 4 Der Body-Mass-Index (BMI) einer Person ist folgendermassen definiert:

$$\text{BMI} = \frac{\text{Masse in kg}}{(\text{Körperlänge in m})^2}$$

Wir nehmen an, dass der BMI in den USA durch die Zufallsvariable $X \sim \mathcal{N}(27, 6^2)$ beschrieben wird. Das nationale Herz-, Lungen- und Blutinstitut definiert die folgenden BMI-Kategorien:

- Untergewicht: BMI ≤ 18.5
- Normalgewicht: $18.5 <$ BMI ≤ 25
- Übergewicht: $25 <$ BMI ≤ 30

- Fettleibigkeit: BMI > 30

(a) Welcher Prozentsatz der Bevölkerung hat Übergewicht?

(b) Welcher Prozentsatz hat Untergewicht oder leidet an Fettleibigkeit?

(c) 5% der Bevölkerung haben einen BMI, der grösser als x ist. Bestimmen Sie x.

Lösung

Gemäss Aufgabenstellung ist $X \sim (\mu = 27, \sigma^2 = 6^2)$ verteilt.

(a) Als Integral aufgeschrieben, sieht die gefragte Wahrscheinlichkeit so aus

$$P(25 < X \leq 30) = \frac{1}{\sigma\sqrt{2\pi}} \int_{25}^{30} e^{-(x-\mu)^2/(2\sigma^2)}\, dx \doteq 0.322 = 32.2\%$$

Mit Python können wir die Lösung folgendermassen berechnen:

```
> stats.norm.cdf(30, 27, 6) - stats.norm.cdf(25, 27,  6)
```

Mit der Tabelle können wir die Lösung folgendermassen berechnen:

$$\begin{aligned} P(25 < X \leq 30) &= P\left(\frac{25-27}{6} < Z < \frac{30-27}{6}\right) \\ &= \Phi(0.5) - \Phi\left(-\frac{1}{3}\right) \\ &= \Phi(0.5) - \left(1 - \Phi\left(\frac{1}{3}\right)\right) \\ &\approx 0.6915 - 1 + 0.6293 \approx 0.3208 \end{aligned}$$

Achtung: ungenau wegen der Tabelle (Rundungsfehler).

(b)

$$P(X < 18.5) + P(X > 30) \approx 0.3868 = 38.68\%$$

Mit Python können wir die Lösung so berechnen:

```
> stats.norm.cdf(18.5, 27, 6) + 1 - stats.norm.cdf(30, 27,  6)
```

Mit der Tabelle können wir die Lösung folgendermassen berechnen:

$$\begin{aligned} P(X < 18.5) + P(X > 30) &= P(X < 18.5) + 1 - P(X \leq 30) \\ &= P\left(Z < \frac{18.5-27}{6}\right) + 1 - P\left(Z < \frac{30-27}{6}\right) \\ &= \Phi\left(-1.41\overline{6}\right) + 1 - \Phi(0.5) \\ &= 1 - \Phi\left(1.41\overline{6}\right) + 1 - \Phi(0.5) \\ &= 2 - 0.92145 - 0.6915 = 0.38705 \end{aligned}$$

Achtung: Ungenauigkeit ist wegen der Tabelle (Rundungsfehler).

(c) Gesucht wird x , so dass

$$P(X < x) = 0.95$$

ist. Die gesuchte Zahl x entspricht dem 0.95-Quantil. Wir erhalten $x \approx 36.869$. Mit Python können wir die Lösung folgendermassen berechnen:

```
> stats.norm.ppf(0.95, 27, 6)
```

Mit der Tabelle gehen wir folgendermassen vor. Wir können direkt ablesen, dass $P(Z \leq 1.64) = 0.9495$ und $P(Z \leq 1.65) = 0.9505$. Also 0.95 kommt nicht genau vor und wir nehmen etwas in der Mitte. D.h wir rechnen $P(Z \leq 1.645) \approx 0.95$, und damit ist das gesuchte z gleich 1.645, weil $1.645 \approx \Phi^{-1}(0.95)$. Jetzt das gesuchte x zu berechnen, ist ganz einfach, nämlich

$$x \quad \approx \quad z \cdot \sigma + \mu = 1.645 \cdot 6 + 27 = 36.87$$