Übungsblatt Nr. 4: in der Übungsstunde

1) a) Simulieren Sie eine Spielserie von 10 Kopf/Zahl Spielen mit einer fairen Münze durch den Aufruf `spielserie(10)` . Wählen Sie das Ergebnis aus einer Liste mit einem Zufallsindex `randint(0,1)` aus.
   b) Was ergibt sich, wenn es mehr als 2 mögliche Spielergebnisse gibt? Passen Sie das Programm an diese Situation an.
   c) Verwenden Sie die Funktion `choice` dafür.
   d) Modellieren Sie damit eine unfaire Münze, die Zahl doppelt so oft als Kopf liefert.

2) a) Erweitern Sie die Aufgabe 1d) um eine Funktion
   `print_hauefigkeiten(ergebnis, ergebnisse)`
   Diese soll für jedes mögliche Ergebnis die Häufigkeit ermitteln, mit der dieses in der Spieleserie aufgetreten ist. Geben Sie die Häufigkeit auch in Prozent an. Spielen Sie hier 1000 Mal.
   b) Schreiben Sie eine eigene Funktion Jedes Ergebnis sollte nur einmal aufgelistet werden.
   c) Geben Sie die Prozent mit genau 2 Nachkommastellen an. Alle Ausgaben sollen schön untereinanderstehen.
   d) Importieren Sie die `matplotlib.pyplot` Funktion und erstellen Sie ein einfaches Balkendiagramm.

Übungsblatt Nr. 4 Hausübung: Die folgenden Aufgaben sind Pflicht und zählen 1 Punkt!

1) a) Erzeugen Sie analog zur Übungsstunde eine Spielserie von 1000 Würfen mit einem fairen Würfel. Jedes Würfeln soll einen `int` 1-6 ergeben. Geben Sie die Häufigkeiten aus und erstellen Sie ein Balkendiagramm.
   b) Verwenden Sie diesmal eine Implementierung, in der Sie jedes Würfeln durch einen String mit einer Ziffer 1-6 modelliert wird. Ihr Würfel sollte die 6 mit Wahrscheinlichkeit 0.5 liefern, alle anderen Zahlen sollten gleich wahrscheinlich sein (unfairer Backgammon-Würfel)

2) Lassen Sie Spieler A gegen Spieler B würfeln. A hat 2 faire Würfel (d.h. 2 Aufrufe des Würfelgenerators), B nur einen. A macht einen Gewinn von $g = 1$, wenn die Summe seiner Augenzahlen echt größer als die Augenzahl von B ist, ansonsten einen Verlust von $g = -5$. Wiederholen Sie das Würfelspiel nun 10000 Mal.
   a) Wie oft hat A gewonnen, wie oft B?
   b) Wer hat am Ende der Spielserie Geld gewonnen?

3) Werten Sie eine Spielserie von `n` Spielen mit Ergebnissen in `[True, False])` aus: Sie wählen einen zufälligen Punkt (x, y) der Ebene aus und gewinnen genau dann wenn $x^2 + y^2 < 1$ gilt (d.h. ihr Punkt liegt im Einheitskreis um den Ursprung. Die Anzahl `n` der Spiele lesen Sie vom User ein!
   a) Generieren Sie die Koordinaten `x, y` des Punktes mit einer Standard-Normalverteilung (in Python mittels eines Aufrufes von `gauss(0., 1.)` . Sie müssen diese Funktion aus `random` importieren.
   b) Generieren Sie die Koordinaten `x, y` des Punktes mit einer Gleichverteilung in [-1,1] (in Python mittels eines Aufrufes von `uniform(-1., 1.)` . Sie müssen diese Funktion aus `random` importieren. Vergleichen Sie die Gewinnhäufigkeit mit der Zahl $\pi/4$ (wenn die Stichprobenanzahl großist! Warum sind diese Werte dann fast gleich?).

4) Monte-Carlo-Integration: Approximieren Sie das Integral $\displaystyle\int_0^1 f(x)\,dx$ für die Funktion $f(x) = e^{-x^2}$. Dieses Integral ist nicht elementar zu berechnen! Wie man leicht sieht, ist der Graph des Integranden im Quadrat $[0, 1] \times [0, 1]$ enthalten. Gesucht ist das Integral, d.h. die Fläche unter der Kurve. Man spielt nun folgendes Spiel: Wähle einen zufälligen gleichverteilten Punkt (x, y) im Einheitsquadrat $[0, 1]\times[0, 1]$. Man gewinnt das Spiel, wenn der Punkt unterhalb der Kurve liegt ($y < f(x)$. Die Gewinnhäufigkeit approximiert die Gewinnwahrscheinlichkeit und diese ist genau gleich dem Verhältnis der 2 Flächen Integral zu Einheitsquadrat. Sie können das Integral mit Maple numerisch berechnen mittels: `evalf(int(exp(-x^2), x=0..1))` ≈ 0.7468241330

Extra-Aufgabe(n): Diese Aufgaben sind freiwillig und zählen 2 Punkte!

5) a) 2D-Random Walk: Ein Betrunkener startet auf einem rechteckigen Gitter von Straßen mit Länge 1 im Punkt $(0, 0)$. Bei jeder Kreuzung geht er zufällig in eine der 4 Himmelsrichtungen bis zur nächsten Kreuzung. Nach 10 solchen Teilstücken landet er bei der Position (x, y) und ist somit $|x| + |y|$ Teilstücke von seiner Ausgangsposition entfernt (d.h.minimal 0, maximal 10). Simulieren Sie diesen Random-Walk 10000 Mal und geben Sie die Häufigkeitstabelle dieses Abstands aus. Welche Abstände werden nie erreicht? Machen Sie auch ein Balkendiagramm.

6) Diesmal interessieren wir uns ausschließlich dafür, ob unser betrunkener Wanderer irgendwann im Lauf seines Weges den Ausgangspunkt (0,0) erneut erreicht (der Start zählt natürlich nicht). Wie oft kommt das bei einem Walk von 20 oder 30 oder 40 Teilstrecken vor?