

Python Aufgabe Pilzsammler:innen Teil 2

Torben Friedrich Görner

Dezember 2022



1 Das Szenario

Täglich gehen Pilzsammler:innen als Gruppe in den Wald, um Pilze zu sammeln. Am Abend wollen alle eine Pilzpfanne kochen. Wenn die Pilzsammler:innen am Abend ihre gesammelten Sorten teilen, muss jeder nur noch 1€ für zusätzliche Pilze ausgeben. Wenn beide nicht teilen, müssen beide 3€ für weitere Pilze ausgeben. Wenn A teilt, B aber nicht, muss A 5€ ausgeben und B 0€. Die Kosten können der Matrix entnommen werden.

Die Pilzsammler:innen gehen jeden Tag los um Pilze zu sammeln und ggf. zu teilen. Welche Strategie zur Entscheidungsfindung ob geteilt werden soll oder nicht ist am erfolgreichsten? Um diese Frage zu untersuchen entwickeln wir eine Simulation.

Matrix

Kostentabelle	A teilt	A teilt nicht
B teilt	1/1	5/0
B teilt nicht	0/5	3/3

Table 1: Kosten für Pilzsammler:innen

Bitte schreibt eure Ergebnisse auf. Nutzt hierfür Word, Google Docs oder ähnliches. Ihr dürft die einzelnen Aufgaben gerne mit eigenen Ideen erweitern.

2 Aufgabe 1 - Beobachtungsfehler

Erstelle eine/n TitForTat Sammler/in ("wie du mir so ich dir" Strategie), wobei der/die Pilzsammler/in mit einer Wahrscheinlichkeit $p = 5\%$ die letzte Handlung seines Gegenübers falsch deutet. In 5 % der Fälle erinnert sich der/die Pilzsammler/in also falsch. Teste deine Implementierung mit einer Gruppe aus 2 Pilzsammler:innen und erhöhe die Anzahl der simulierten Tage auf *1.000.000*.

3 Aufgabe 2 - Gruppengröße

Macht es einen Unterschied wenn wir die Gruppengröße erhöhen? Wie ändern sich die mittleren Kosten der Pilzsammler:innen?

4 Aufgabe 3 - Andere Beobachtungsfehler

Was passiert, wenn wir den Beobachtungsfehler auf 25 %, 50 %, 75 % oder 95% setzen? Wie ändern sich die mittleren Kosten der Pilzsammler:innen?

5 Aufgabe 4 - Verbesserungen

Welche Möglichkeiten fallen dir ein, um das System zu verbessern?

6 Tipp

Für die Erzeugung einer Zufallszahl in Python kannst du *randrange* aus *random* nutzen. Die Funktion *randrange* bekommt jeweils einen Parameter für die obere und die untere Grenze. Im folgenden Beispiel nimmt *r* einen Wert zwischen 0 und 42 an.

```
from random import randrange  
  
r = randrange(0, 42)
```