

Übungsblatt 2 – Angabe

Aufgaben 2.Einheit

- Die Jupyter Notebooks der gelösten Aufgaben bitte im angelegten Übungsblatt Ordner speichern.
- Bitte erstellen sie für jede gelöste Aufgabe ein eigene/s Datei/Notebook (Bsp1, Bsp2, ...).
- Komprimieren sie den Ordner Uebungsblatt_x_se201xxx (zip) und laden ihn bei der Abgabe am eCampus hoch.

Nicht ordnungsgemäße oder verspätete Abgaben können nicht zur Beurteilung herangezogen werden

1. Programmieren Sie Lotto 6 aus 45 (ohne Zusatzzahl) wie folgt - kopieren Sie sich dafür das Grundgerüst *lotto_grundgeruest.ipynb*:

Hinweis: Versuchen Sie den Code in Funktionen aufzuteilen und diese aufzurufen.

- a) Eingabe: Das Programm soll vom Benutzer einen Lottotipp entgegennehmen. Hierfür sollen sie sechs Zahlen per Standardeingabe in eine Liste einlesen. Stellen Sie sicher, dass nur Zahlen zwischen 1 und 45 erlaubt sind und keine Zahl doppelt verwendet werden darf. Dieser Punkt ist im Grundgerüst teilweise enthalten.
- b) Überprüfen Sie die Eingabe auf Zahlen (z.B. `isNumeric` oder mittels `try/except`), bei einer falschen Eingabe (zu viele, zu wenige oder doppelte Zahlen) soll der User erneut nach den Zahlen gefragt werden. Die Eingabe können Sie mit `.split()` nach Leerzeichen in eine Liste splitten somit haben Sie alle Eingaben wie folgt [20, 17, 15, 10, 1, 45].
- c) Lesen Sie danach die Anzahl der Ziehungen (min. 10) ein. Dabei soll wieder überprüft werden, ob eine Zahl eingegeben wird und die Zahl mindestens ≥ 10 (Ziehungen) ist
- d) Ziehungen der Computerzahlen (ohne Zusatzzahl): Bei einer Ziehung werden insgesamt 6 Zahlen (Computerzahlen) zufällig im Bereich 1-45 gezogen. Stellen sie sicher, dass keine Zahl doppelt gezogen wird. Es wird für die Anzahl der Ziehungen eine Schleife benötigt, die so lange läuft, bis alle Ziehungen durch sind. Überlegen Sie sich, welche Möglichkeiten es gibt, so eine Ziehung zu realisieren.
- e) Schreiben Sie eine Vergleichsfunktion bei der Sie für jede Ziehung die Anzahl der gleichen Zahlen zurückbekommen.
Code: `len(set(list_computer_numbers).intersection(list_user_numbers))`
Hinweis: Eine Erklärung zu dieser Codezeile finden Sie am eCampus unter „Unterlagen aus der Übung 2 - Girsule/Lotto.ipynb“.
- f) Vergleichen Sie jede Ziehung (= zufällig gezogenen „Kugeln“) mit dem abgegebenen Tipp. Stellen sie fest, ob es sich um einen 0er (keine richtig), 1er, 2er, 3er, 4er, 5er oder Lotto-6er handelt. Speichern Sie das Ergebnis aus e), wie viele der eingegebenen Zahlen mit den Computerzahlen übereinstimmen in einem Dictionary oder einer Liste ab. Beispiel eines Dictionaries mit der Anzahl der richtigen Zahlen als Key und den Counter als Value, könnte wie folgt aussehen:
`gewinn = {0: 0, 1: 0, 2: 0, 3: 0, 4: 0, 5: 0, 6: 0}`

Möglicher Output:

```
tip input: 20 17 15 10 1 45
draw count: 150
0er: 101
1er: 20
2er: 10
3er: 9
4er: 8
5er: 2
6er: 0
```

Punkte: 15

Übungsblatt 2 – Angabe

2. Visualisierung und Analyse der Lotto-Ziehungen

In dieser Aufgabe sollen Sie die Häufigkeit und Verteilung der Ergebnisse aus mehreren Lotto-Ziehungen visualisieren. Die Visualisierungen und Berechnungen geben Aufschluss darüber, wie oft eine bestimmte Anzahl an Übereinstimmungen (0er bis 6er) erreicht wurde, wie wahrscheinlich ein bestimmter Treffer ist und was der erwartete, „durchschnittliche“ Treffer ist.

Falls Sie die erste Aufgabe bereits gelöst haben, verwenden Sie bitte die Ergebnisse (Dictionary, Liste etc. mit der Anzahl der Übereinstimmungen pro Ziehung) für die folgenden Berechnungen und Visualisierungen. Falls nicht, steht ein Beispiel-Datensatz bereit, den Sie stattdessen verwenden können:

`gewinn = {0: 98, 1: 25, 2: 15, 3: 8, 4: 3, 5: 1, 6: 0}`

Häufigkeitsverteilung der Übereinstimmungen

- Erstellen Sie ein Balkendiagramm, das die Häufigkeit der Übereinstimmungen (0er bis 6er) darstellt.
- Verwenden Sie entweder das `gewinn`-Dictionary (oder Datentyp nach Wahl) aus Aufgabe 1 (falls vorhanden) oder den bereitgestellten Beispiel-Datensatz.
- **Visualisierung:** Plotten Sie die Anzahl der Übereinstimmungen (X-Achse) und deren Häufigkeit (Y-Achse) als Balkendiagramm.

Hinweise zur Umsetzung: Fügen Sie Titel und Achsenbeschriftungen zu allen Diagrammen hinzu, um die Darstellung klar und verständlich zu machen.

Punkte: 20

Übungsblatt 2 – Angabe

3. Nehmen Sie wieder ihr Lotto – Beispiel Bsp1 als Vorlage/Grundgerüst. Programmieren Sie das Beispiel wie folgt um:
- a) Nachdem Sie ihre Lottozahlen eingegeben haben, speichern Sie sich die Startzeit (siehe Punkt c)
 - b) Sie entfernen die Abfrage „wie viele Ziehungen gemacht werden“ und lassen das Programm so lange laufen, bis ein Lotto 5er erreicht wird (Lotto 6er würde zu lange dauern, ca. 1-10min).
 - c) Es soll ein Timer mitlaufen wie lange es braucht einen Lotto 5er zu erreichen, d.h. nachdem Sie ihre Lottozahlen eingegeben haben, speichern Sie sich die Startzeit (`startTime = time.time()`) und sobald ein Lotto 5er erreicht wurde, wird die Zeit gestoppt und die Dauer ausgerechnet (`duration = time.time() - startTime`).

Hinweis: um `time` verwenden zu können, muss die Klasse „`time`“ importiert werden - `import time`.

Möglicher Output:

Meine Zahlen: [10, 20, 3, 17, 7, 12]
Ziehungen: 194598
Time: 2.1808834075927734 sec

Punkte 15

Gesamt 50 Punkte