

Simulation mit GeoGebra 6: n-faches Würfeln mit einem Hexaeder

Stell dir folgende Situation vor: deine Mathelehrerin gibt dir als Hausaufgabe auf, mit einem Spielwürfel 100mal zu würfeln, die Ergebnisse zu notieren und auszuwerten. Die Auswertung soll ein Säulendiagramm sowie die Berechnung des arithmetischen Mittels und der empirischen Standardabweichung umfassen.

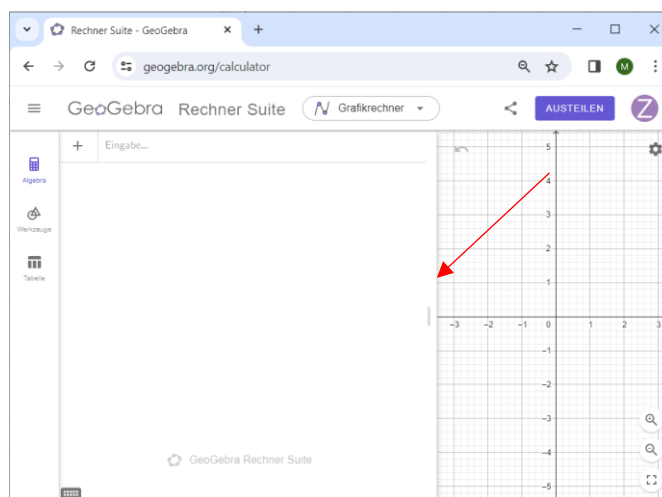
Dein Problem: du hast gerade keinen Spielwürfel zur Hand oder du hast zu wenig Zeit, um 100mal zu würfeln oder du bist nicht sonderlich motiviert dazu...

Eine Möglichkeit: du denkst dir die Daten aus und hast ein kleines bisschen ein schlechtes Gewissen, dass du beim Anfertigen der Hausaufgabe geschummelt hast.

Andere Möglichkeit: du verwendest GeoGebra...

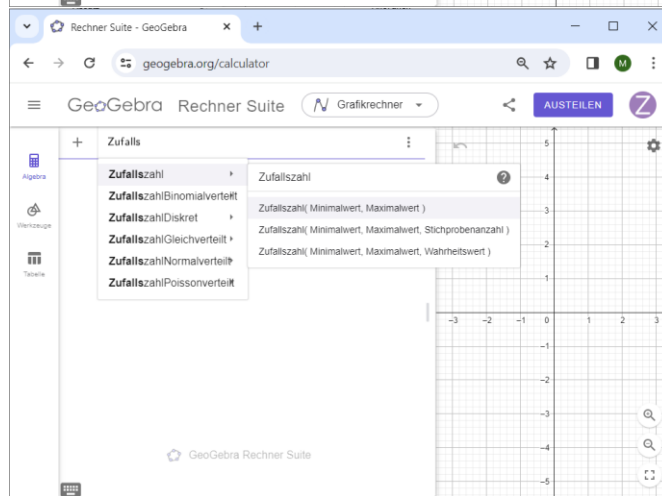
Hier folgt nun die Anleitung für die zweite Möglichkeit:

1. Starte GeoGebra 6 (lokal installiert oder [online](#)).
Tipp: ziehe an der senkrechten Unterteilungslinie, um das Eingabefenster zu vergrößern.



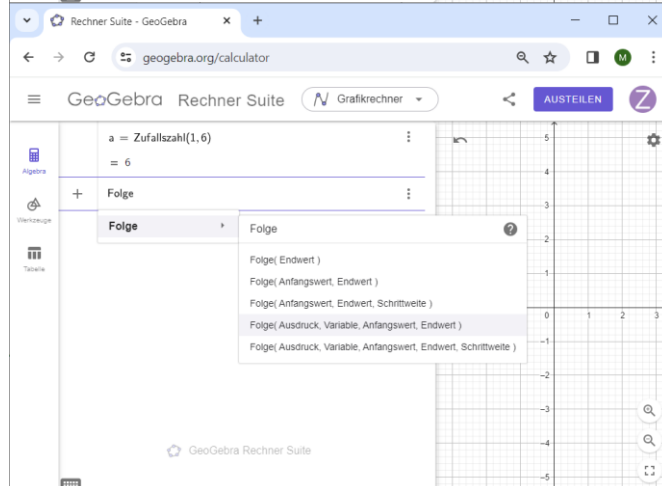
2. Mit GeoGebra hat man unterschiedliche Möglichkeit, Zufallszahlen zu generieren. Möchte man das Würfeln mit einem sechsseitigen Würfel („Hexaeder“) simulieren, verwendet man am besten den Befehl `Zufallszahl(1,6)`.

Es genügt, lediglich „Zufalls“ in das Eingabefeld zu schreiben, da man dann bereits den korrekten Befehl vom Programm vorgeschlagen bekommt.



3. Möchte man nun gleichzeitig 100 Würfe durchführen, legt man eine *Folge* an. Der Befehl lautet dann:
`Folge(Zufallszahl(1,6),i,1,100)`

(Der Befehl bewirkt, dass hintereinander 100mal eine Zufallszahl aus dem Bereich zwischen 1 und 6 erzeugt und in eine Liste geschrieben wird.)



4. Wir möchten nun auszählen, wie häufig die Augenzahl 1, Augenzahl 2, usw. geworfen worden ist. Dafür definieren wir zunächst eine Liste, die als Klasseneinteilung fungieren wird. Sie muss daher die Werte von 1 bis 7 (nicht nur 6!!!) beinhalten. Die Klassenbereiche sind dann die Intervalle [1;2), [2;3),..., [5;6) und [6;7).

(Alle Intervalle schließen den linken Wert ein. Alle bis auf das letzte schließen den rechten Wert aus. In der hier verwendeten Schreibweise erkennt man das an der unterschiedlichen Art der Klammern. Runde Klammer: Wert wird ausgeschlossen, eckige Klammer: Wert wird eingeschlossen.)

5. Mit dem Befehl *Häufigkeit* kann man nun die Anzahl der unterschiedlichen Ergebnisse beim Würfeln auszählen lassen. Der Befehl lautet: Häufigkeit(12,11)

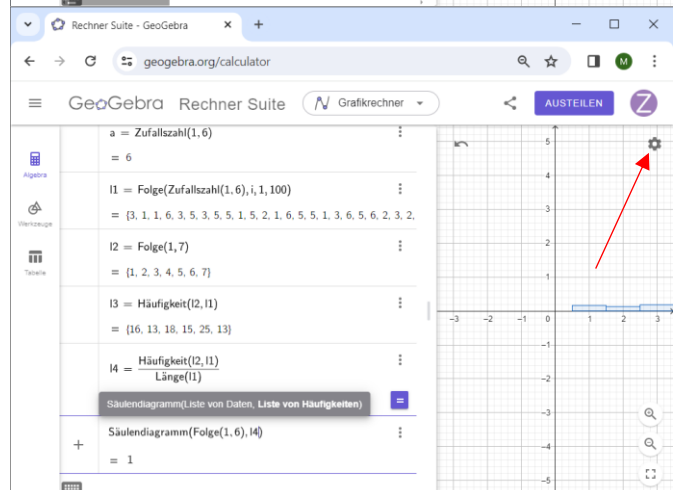
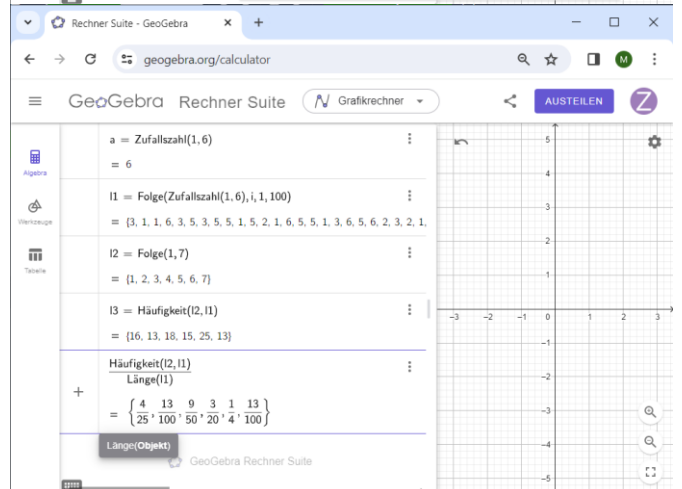
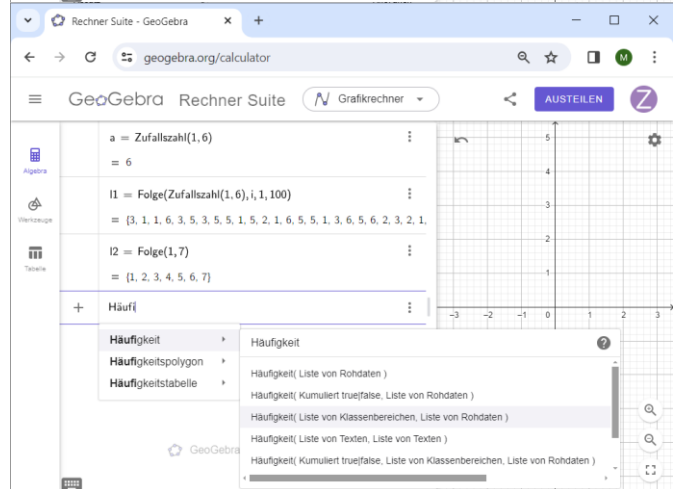
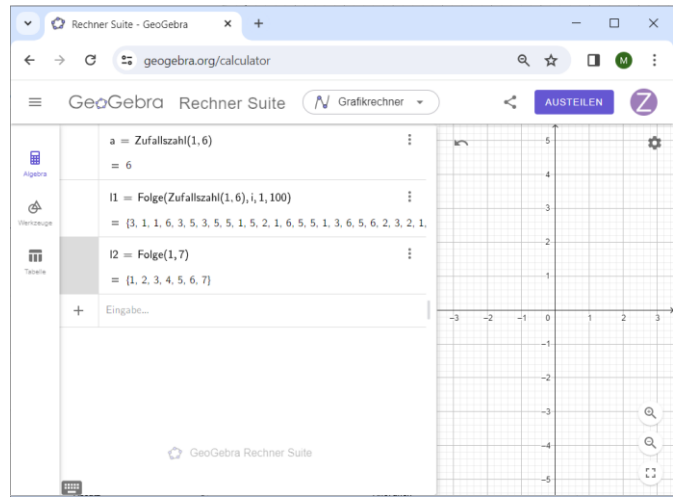
(Achtung: gemeint ist ein kleines „L“, kein großes „i“!!!)

Man erhält eine Liste mit 6 Einträgen, die angeben, wie häufig die Augenzahl 1, Augenzahl 2 usw. gewürfelt worden ist.

6. Wenn man anstelle der absoluten Häufigkeiten lieber die relativen Häufigkeiten in der Liste angeben möchte, dividiert man den Ausdruck noch durch die Anzahl der Würfe. Dafür kann man den Befehl *Länge* verwenden. (Alternativ könnte man auch direkt durch 100 dividieren, aber später -im letzten Teil der Anleitung- wird es sich als vorteilhaft erweisen, wenn man hier allgemein formuliert.)

7. Mit dem Befehl *Säulendiagramm* kann man die erzeugten Daten nun grafisch ausgeben. Achtung: die *Liste von Daten* umfasst genau die 6 Werte, die beim Würfeln erreicht werden können, also eine Liste mit natürlichen Zahlen von 1 bis 6.

Um die Darstellung des Diagramms anzupassen, muss man in die Einstellungen (kleines Zahnrad oben rechts) gehen.



8. Die x-Achse sollte Werte zwischen -1 und 7 umfassen, auf der y-Achse sollten es Werte zwischen -0,1 und 1 sein. Anschließend das Einstellungsfenster durch Anklicken des x schließen. Das Fenster ggf. anschließend mit der Maus noch zurechtrücken.

9. Anschließend den *Mittelwert* berechnen: Mittel(l1)

10. Und die *Standardabweichung*: stdevp(l1)

11. Mit dem Befehl *AktualisiereKonstruktion()* werden alle Zufallszahlen neu erzeugt. Entsprechend wird damit dann auch das Säulendiagramm angepasst und das arithmetische Mittel sowie die empirische Standardabweichung werden neu berechnet.

Falls man mehr als 100mal würfeln soll, kann man leicht eine Anpassung vornehmen: einfach im Befehl für die Liste l1 nicht bis 100 summieren, sondern bis zur gewünschten (anderen) Zahl, z.B. bis 1000 oder 5000 oder 10000 ... (Die Anpassung funktioniert aber nur, wenn man in 6. als Divisor den Befehl *Länge* verwendet hat!)

