Übungsblatt 6

Aufgabe 1a

Bei sowohl 6 als auch 7 Würfen stimmen die Resultate überein:

6 Würfe: Erwartungswert = 3.0 & Varianz = 1.5

7 Würfe: E(X)= 3.5 & V(X) = 1.75

Aufgabe 1b

Zunächst werden wie immer die „itertools“ (Sammlung von nützlichen Python Befehlen) eingeführt um diese verwenden zu können. Der Ergebnisraum Ω soll den kartesischen Produkten von “K” und „Z“ entsprechen und insgesamt wird (in unserem Fall zumindest) 5 mal geworfen. Der Erwartungswert E und die Varianz V werden zunächst auf = 0 gesetzt. Danach wird bzgl. des Erwartungswertes festgelegt für alle ω im Ergebnisraum, dass sich der Erwartungswert als „Erwartungswert + die Summe der Anzahl der Kopfwürfe“ zusammensetzt. Dieser „günstige“ Erwartungswert muss dann wie immer noch durch alle möglichen Ausgänge dividiert werden (E = E / len(Ω)). Bzgl. der Varianz wird für alle ω im Ergebnisraum festgelegt, dass sich die Varianz als „Varianz + (Summe der Anzahl der Kopfwürfe minus Erwartungswert) zum Quadrat“ zusammensetzt. Danach wird erneut durch alle möglichen Ausgänge dividiert. Zum Schluss wird lediglich festgelegt, dass „E“ als Erwartungswert und „V“ als Varianz geschrieben werden soll.

Inhaltlich weisen die zwei Programme minimale Unterschiede auf: beim zweiten Programm wird der Befehl „ω.count("K")“ als „x“ definiert. Somit bleibt es nachher erspart „ω.count("K")“ jedes Mal auszuschreiben, sondern man kann einfacherweise „x“ schreiben. Ein weiterer Unterschied ist auch, dass beim zweiten Programm die Varianz zunächst so berechnet wird: V = V + x*x → das heißt, dass das E*E noch nicht gleich abgezogen wird, dass erfolgt dann erst später bei der finalen Division V = V / len(Ω) - E*E.

Aufgabe 2a

Das Programm berechnet jeweils die Wahrscheinlichkeit dafür, dass „Kopf“ 0 mal, 1 mal, 2 mal, ..., bis 5 mal geworfen wird. 2 oder 3 Kopfwürfe sind demnach mit einer Wahrscheinlichkeit von 31.25 % am ehesten zu erwarten.

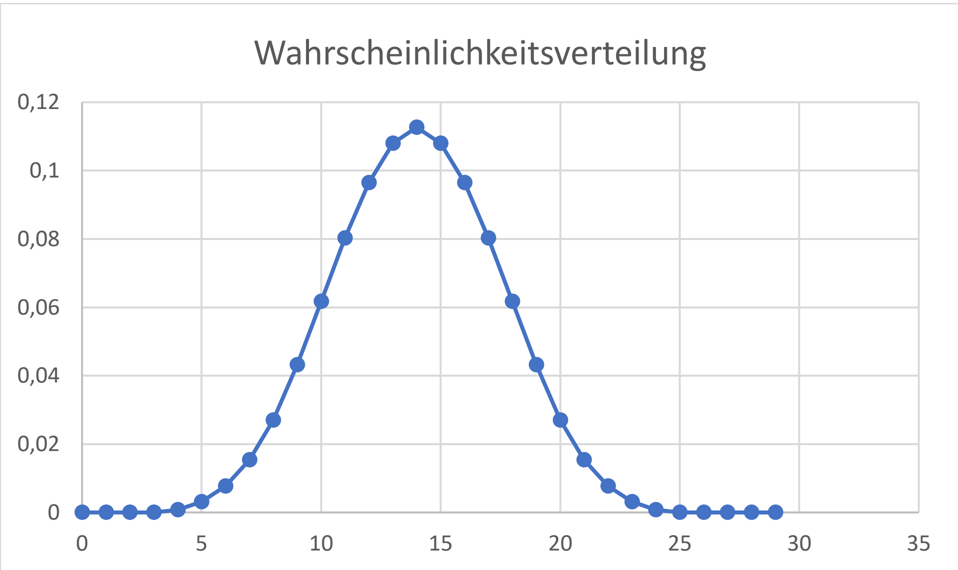
Aufgabe 2b

```python
import itertools
Ω = set(itertools.product({1, 2, 3, 4, 5, 6}, repeat=4))
mögliche_Werte = range(30)
# Zu Beginn hat jeder mögliche Wert der Zufallsvariablen
# eine Häufigkeit von 0.
Anzahl_Köpfe = [0]*len(mögliche_Werte)
for ω in Ω:
    Anzahl_Köpfe[sum(ω)] += 1
Wahrscheinlichkeiten = [h/len(Ω) for h in Anzahl_Köpfe]
print(list(zip(*(mögliche_Werte, Wahrscheinlichkeiten))))
```

```
[(0, 0.0), (1, 0.0), (2, 0.0), (3, 0.0), (4, 0.000771604938271
49), (5, 0.0030864197530864196), (6, 0.007716049382716049), (7,
0.015432098765432098), (8, 0.02700617283950617), (9, 0.043209876
543209874), (10, 0.06172839506172839), (11, 0.08024691358024691)
, (12, 0.09645061728395062), (13, 0.10802469135802469), (14, 0.1
1265432098765432), (15, 0.10802469135802469), (16, 0.09645061728
395062), (17, 0.08024691358024691), (18, 0.06172839506172839), (
19, 0.043209876543209874), (20, 0.02700617283950617), (21, 0.015
432098765432098), (22, 0.007716049382716049), (23, 0.00308641975
30864196), (24, 0.0007716049382716049), (25, 0.0), (26, 0.0), (2
7, 0.0), (28, 0.0), (29, 0.0)]
```

Hier wird die Wahrscheinlichkeit für die Summe der Augenzahlen eines Würfels, der 4 mal geworfen wird, berechnet. Nach 4 maligem Würfeln werden die jeweiligen Augenzahlen also summiert und die Wahrscheinlichkeit für jeden möglichen Augensummen-Wert berechnet. Das Ergebnis zeigt uns, dass die Augensummen 0, 1, 2, 3, 25, 26, 27, 28, 29 nicht vorkommen können und, dass eine Augensumme von 14 mit einer Wahrscheinlichkeit von 11,27% am wahrscheinlichsten ist.

Aufgabe 2c



So sieht bei mir das Ergebnis aus. Die gesamte Excel Datei findet sich im Mail Anhang ebenso. Aus der Grafik wird gut ersichtlich, dass es sich um eine Normalverteilung handelt.

<u>Aufgabe 2d</u>

```python
import itertools
Ω = set(itertools.product({1, 2, 3, 4, 5, 6}, repeat=4))
mögliche_Werte = range(30)
# Zu Beginn hat jeder mögliche Wert der Zufallsvariablen
# eine Häufigkeit von 0.
Anzahl_Köpfe = [0]*len(mögliche_Werte)
for ω in Ω:
    Anzahl_Köpfe[sum(ω)] += 1
Wahrscheinlichkeiten = [h/len(Ω) for h in Anzahl_Köpfe]
for w in zip(*(mögliche_Werte, Wahrscheinlichkeiten)):
    print(f"{w[0]},{w[1]}")
Verteilung = [sum(Wahrscheinlichkeiten[:h]) for h in
    mögliche_Werte]
print(list(zip(*(mögliche_Werte, Verteilung))))
```

```
12,0.09645061728395062
13,0.10802469135802469
14,0.11265432098765432
15,0.10802469135802469
16,0.09645061728395062
17,0.08024691358024691
18,0.06172839506172839
19,0.043209876543209874
20,0.02700617283950617
21,0.015432098765432098
22,0.007716049382716049
23,0.0030864197530864196
24,0.0007716049382716049
25,0.0
26,0.0
27,0.0
28,0.0
29,0.0
[(0, 0), (1, 0.0), (2, 0.0), (3, 0.0), (4, 0.0), (5, 0.000771604
9382716049), (6, 0.0038580246913580245), (7, 0.011574074074074074
3), (8, 0.02700617283950617), (9, 0.05401234567901234), (10, 0.0
9722222222222221), (11, 0.1589506172839506), (12, 0.239197530864
1975), (13, 0.3356481481481814), (14, 0.4436728395061728), (15,
 0.5563271604938271), (16, 0.6643518518518519), (17, 0.760802469
1358025), (18, 0.8410493827160495), (19, 0.9027777777777779), (2
0, 0.9459876543209877), (21, 0.9729938271604939), (22, 0.9884259
259259259), (23, 0.996141975308642), (24, 0.9992283950617284), (
25, 1.0), (26, 1.0), (27, 1.0), (28, 1.0), (29, 1.0)]
```

Hier wird berechnet, wie hoch die Wahrscheinlichkeit ist, dass die Summe der gewürfelten Augenzahlen kleiner ist als diese Augenzahl ist. Bei der Berechnung für 11 wird also ermittelt wie wahrscheinlich es ist, dass die Augenzahlensumme von 4 Würfen kleiner als 11 ist. Da die kleinstmögliche Augenzahlensumme 4 ist, kommt bei den ersten 4 Werten natürlich 0 heraus, da es nicht kleiner werden kann. Dasselbe gilt in umgekehrter Weise für alle Werte ab 25: Die größtmögliche Augenzahlensumme ist 24, also kommt bei den Werten ab 25 eine Wahrscheinlichkeit von 100% heraus, dass die Summe kleiner ist.