23.5 Binomialverteilung

Definition 23.5.1 Bernoulli-Versuch

Ein Zufallsversuch mit nur zwei Ergebnissen heisst ein Bernoulli-Versuch. Es hat sich eingebürgert, eine der beiden Möglichkeiten «Erfolg», und die andere «Misserfolg» zu nennen. Die Wahrscheinlichkeit, für den Erfolg wird mit p, die Wahrscheinlichkeit für den Misserfolg mit q bezeichnet. Weil es nur diese beiden Möglichkeiten gibt, gilt $q = 1 - p$.

Beispiele:

- Würfeln mit einem Würfel. Erfolg bedeutet eine 6 Würfeln. $p = \frac{1}{6}$, $q = \frac{5}{6}$.
- Beim Roulette auf Rot setzen. $p = \frac{18}{37}$, $q = \frac{19}{37}$.
- Münzwurf, Erfolg ist «Kopf». $p = \frac{1}{2}$, $q = \frac{1}{2}$.

Definition 23.5.2 Binomialverteilte Zufallsvariable

Ein Bernoulli-Experiment wird n mal wiederholt. Sei X die Zufallsvariable, die die Anzahl Erfolge zählt. X folgt der Binomialverteilung mit Parametern n und p, oft mit $X \sim \mathrm{Bin}(n, p)$ notiert. Es gilt

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad = \binom{n}{k} p^k q^{n-k}$$

notiert.

Aufgabe 23.19 Leiten Sie die Formel für $P(X = k)$ für eine binomialverteilte Zufallsvariable X mit Parametern n und p her. Zeichnen Sie evtl. einen Baum des Zufallversuchs.

Merke 23.5.3 Erwartungswert einer binomialverteilten Zufallsvariablen

Sei $X \sim \mathrm{Bin}(n, p)$. Dann ist der Erwartungswert (Durchschnitt)

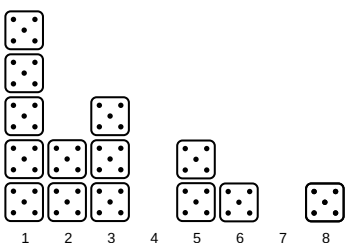
$$\mathrm{E}(X) = np$$

und die Varianz (Quadrat der Standardabweichung)

$$\mathrm{Var}(X) = np(1-p)$$

Aufgabe 23.20

Im Technorama gibt es folgenden Versuch: Man startet mit $n = 60$ Würfeln. In jedem Schritt (mit Nummer i) wird mit allen noch verbleibenden Würfeln gewürfelt und alle Fünfer aussortiert und vertikal aufeinander auf Position i gestapelt. Dies wird so lange wiederholt, bis alle Würfel gestapelt sind. Das Resultat könnte etwa wie im Bild rechts aussehen.



Berechnen Sie,

a) wie hoch der Würfelturm beim ersten Wurf durchschnittlich wird.

b) wie gross die Wahrscheinlichkeit ist, beim ersten Wurf einen Turm von 5, 10 oder 20 Würfeln auszusortieren.

c) Schätzen Sie ein 95%-Intervall ab für die Höhe vom ersten Turm.

d) Simulieren Sie diesen Versuch in Python und ermitteln Sie näherungsweise Erwartungswert und Standardabweichung der Anzahl nötigen Schritte, bis alle Würfel aussortiert sind.

Studieren Sie folgende Python-Funktionen und vervollständigen Sie das Programm bei den mit `TODO` markierten Stellen.

```
from random import randrange

def wuerfeln(n):
    k = 0                       # Zähler für Anzahl Fünfen
    for i in range(n):          # n mal wiederholen
        if randrange(6)==0:     # Ganzzahlige Zufallszahl von 0 bis und mit 5
            k+=1
    return k

def versuch(n):
    schritte=0                  # Zähler für Anzahl Schritte
    while n>0:                  # Solange noch Würfel übrig sind
        n = n - wuerfeln(n)     # Würfel herauslegen
        schritte+=1
    return schritte

anzahlVersuche = 10000
wuerfel = 60            # Anzahl Würfel für einen Versuch
werte = []              # Liste mit Werten

# Liste füllen
for i in range(anzahlVersuche):
    werte.append(versuch(wuerfel))

summe = sum(werte)  # Summe aller Werte
schnitt = # TODO: Durchschnitt aller Werte berechnen

# Variable zur Berechnung der Standardabweichung
sigma = 0
for i in range(anzahlVersuche):
    # werte[i]  ist der Wert an Stelle i
    # Standardabweichung aufsummieren
    sigma = sigma + # TODO: Quadrierte Abweichung vom werte[i] zu schnitt

# TODO: summe dividideren, dann Wurzel ziehen
sigma = # TODO

sigmaxbar = sigma/(anzahlVersuche**0.5)
print(f"Schnitt: {schnitt}, StdDev: {sigma}, StdDev vom Schnitt: #{sigmaxbar} ")
```

e) Berechnen Sie den exakten Erwartungswert für die noch nötigen Schritte, wenn nur noch ein Würfel übrigbleibt. Sei dazu E_1 dieser Erwartungswert. Mit diesem lässt sich folgende Gleichung aufstellen:

$$E_1 = p \cdot 1 + q \cdot (1 + E_1)$$

Erklären Sie die Gleichung und lösen Sie nach E_1 auf. Erscheint Ihnen das Resultat plausibel?

f) Finden Sie eine Gleichung für den exakten Erwartungswert E_2 für die noch nötigen Schritte, wenn noch zwei Würfel übrigbleiben. Darin kommt auch der Erwartungswert E_1 für nur einen Würfel vor.

g*) Schreiben Sie analog dazu eine Gleichung für E_k auf, die erwarteten Anzahl Schritte, wenn noch k Würfel übrigbleiben. Diese soll ein Summenzeichen $\sum$, $P(X = i)$ und E_i enthalten. Lösen Sie dann diese Gleichung nach E_k auf und fassen Sie zusammen.

h) Studieren und erklären Sie folgende Python-Funktionen:

```
# Binomialkoeffizient n tief k
def binom(n,k):
    b = 1
    if k>n/2:
        k=n-k
    for i in range(k):
        b = b * (n-i)/(k-i)
    return b


# Wahrscheinlichkeit, dass eine binomialverteilte Zufallsvariable mit Parametern
# (n,p) den Wert k annimmt.
def binprob(n,k,p):
    return binom(n,k)*p**k*(1-p)**(n-k)
```

i) Programmieren Sie die in g) bestimmte Formel und vergleichen Sie die Resultate mit jenen der Aufgabe d).

```
# Liste mit Werten für E_k, start bei E_0 = 0
e = [0]
for k in range(1,61):   # Berechnung für k von 1 bis und mit 60
    s=1
    # Summe berechnen
    for i in range(1,k):  # Summe von 1 bis und mit k-1
        # e[k-i] ist der bereits berechnete Erwartungswert für k-i Würfel
```

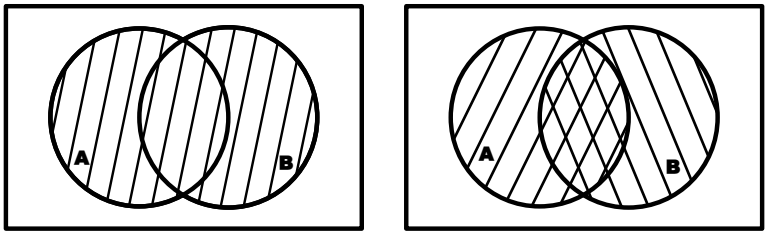
```
        s = s + # TODO: Hier den Ausdruck im Summanden hinzufuegen

    s = s / # TODO: Summe teilen
    # Berechneten Wert der Liste der Erwartungswerte hinzufügen
    e.append(s)
```

j) Nehmen wir an, wir hätten sehr viele Würfel am Anfang, z.B. $n = 1'000'000$. Die obigen Methoden zur Berechnung sind für solche grosse Zahlen nicht wirklich praktikabel. Schätzen Sie die durchschnittliche Anzahl benötigter Würfe ab.

k) Berechnen Sie im Python-Programm die absolute Abweichung der exakten Werte von der Schätzung. Was stellen Sie fest? Wie erklären Sie sich diesen Sachverhalt? Bestimmen Sie damit eine Formel für eine hoffentlich genauere Schätzung.

Um den Logarithmus zu berechnen ist "import math" nötig:

```
import math
schaetzung = math.log(k)/math.log(6/5)
```

d)


Links ist die Vereinigungsmenge. Rechts wird die Schnittmenge doppelt gezählt, muss also wieder abgezogen werden.

Lösung zu 23.18 ex-repe-hivtest

Gesucht ist die Wahrscheinlichkeit $P(\text{gesund} \mid \text{pos. Test})$.

Die Wahrscheinlichkeit, dass ein Test positiv ist kann wie folgt berechnet werden. Zeichnen Sie zum besseren Verständnis den entsprechenden Baum.

$P(\text{pos. Test}) = P(\text{gesund } \textbf{und} \text{ pos. Test}) + P(\text{infiziert } \textbf{und} \text{ pos. Test}) = P(\text{gesund}) \cdot P(\text{pos. Test} \mid \text{gesund}) + P(\text{infiziert}) \cdot P(\text{infiziert} \mid \text{pos. Test}) = 0.9975 \cdot 0.002 + 0.0025 \cdot 0.99 = 0.00447$

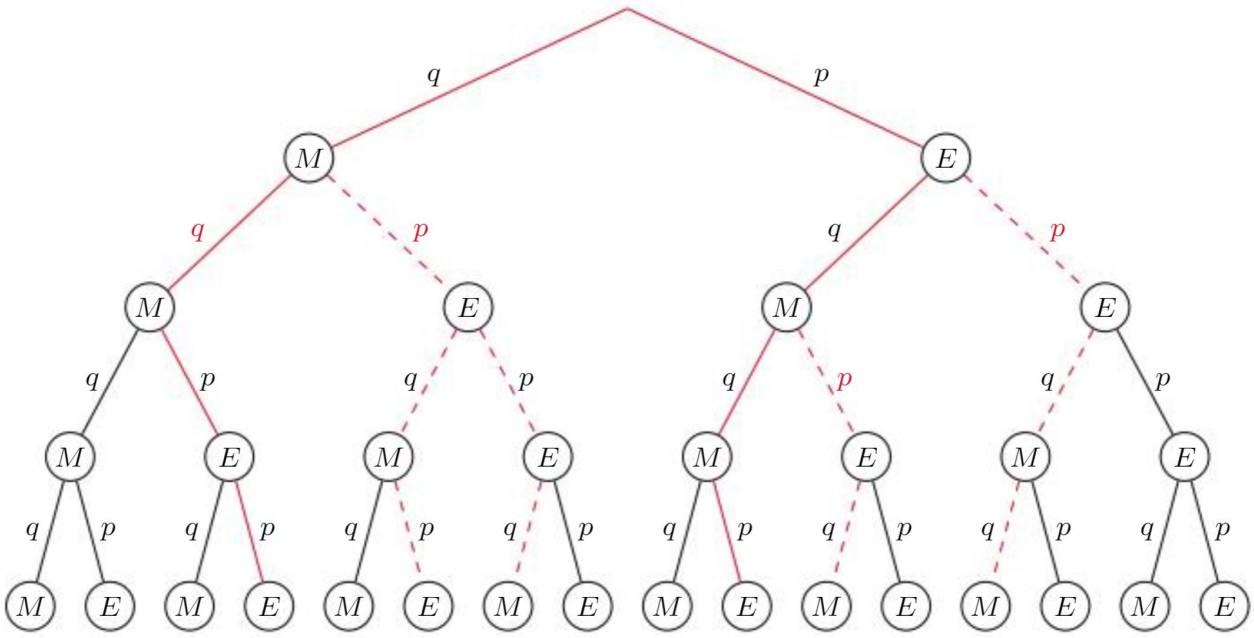
Die gesuchte Wahrscheinlichkeit ist also

$P(\text{gesund} \mid \text{pos. Test}) = \frac{P(\text{gesund } \textbf{und} \text{ pos. Test})}{P(\text{pos. Test})} = \frac{0.9975 \cdot 0.002}{0.00447} \approx 0.4463$. Konkret heisst das, wenn eine zufällige Person einen positiven Test erhält, beträgt die Wahrscheinlichkeit knapp 45%, dass die Person gar nicht infiziert ist. Darum ist der Test nicht ohne ärztliche Begleitung zugänglich.

Der Test ist dann aber brauchbar, wenn die Person nicht zufällig ausgewählt wird, sondern bereits ein erhöhtes Risiko mitbringt, weil sie z.B. ungeschützte sexuelle Kontakte mit Personen aus Risikogruppen oder gar mit infizierten Personen hatte.

Lösung zu 23.19 ex-binomalverteilung-herleiten

Alle möglichen Ausgänge der n wiederholten Bernoulli-Versuche können als Baum dargestellt werden, hier am Beispiel $n = 4$:



Die Zufallsvariable X zählt nur, wie viel mal ein Erfolg stattgefunden hat. Die konkrete Reihenfolge ist aber nicht wichtig.

Wenn $X = k$ haben wir k Erfolge «E» und $(n - k)$ Misserfolge «M». Ein Versuch kann also Wort der Länge n aus den Buchstaben «E» und «M» aufgefasst werden. Um die k Buchstaben «E» auf die n möglichen Plätze zu verteilen, gibt es $\binom{n}{k}$ Möglichkeiten.

Jede dieser Möglichkeiten hat die Wahrscheinlichkeit $p^k(1-p)^{n-k}$. Daher

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Lösung zu 23.20 ex-technoram-wuerfel-aussortieren

a) $E(X) = np = 10$.

b) $P(X = 5) \approx 0.031016$, $P(X = 10) \approx 0.137013$ $P(X = 20) \approx 7.80064 \cdot 10^{-4}$.

c) $\sigma = \sqrt{np(1-p)} \approx 2.88675$. Vertrauensintervall $10 \pm 2 * \sigma : [4, 16]$.

d) Wiederholt man den Versuch 10'000 mal erhält man z.B. Schnitt: 26.1591, $\sigma_{\overline{x}} : 0.071$

Mit 100'000 Versuchen wurde Schnitt: 26.19816, $\sigma_{\overline{x}} : 0.0221$ erhalten.

```
from random import randrange

def wuerfeln(n):
    k = 0
    for i in range(n):
        if randrange(6)==0:
            k+=1
    return k

def versuch(n):
    schritte=0
    while n>0:
        n = n - wuerfeln(n)
        schritte+=1
    return schritte



anzahlVersuche = 10000
wuerfel = 60            # Anzahl Würfel für einen Versuch
werte = []              # Liste mit Werten

# Liste füllen
for i in range(anzahlVersuche):
    werte.append(versuch(wuerfel))

summe = sum(werte)  # Summe aller Werte
schnitt = sum(werte)/anzahlVersuche

abweichungen = []
sigma = 0
for i in range(anzahlVersuche):
    # werte[i]  ist der Wert an Stelle i
    # Standardabweichung aufsummieren
    sigma = sigma + (werte[i]-schnitt)**2

sigma = (sigma/(anzahlVersuche-1))**0.5
sigmaSchnitt = sigma/anzahlVersuche**0.5
print(f"{anzahlVersuche} Versuche mit {wuerfel} Würfeln")
print(f"Schnitt: {schnitt}, stdDev: {sigma}, stdDev vom Schnitt: {sigmaSchnitt}")
```

e) Ist noch ein Würfel übrig, gibt es beim ersten Schritt zwei Möglichkeiten. Entweder man würfelt eine 5 und ist damit nach einem Schritt fertig, oder man würfelt keine 5, hat einen Schritt verbraucht, und braucht danach im Mittel noch E_1 Schritte. D.h. mit W'keit $p = \frac{1}{6}$ braucht man einen Schritt, mit W'keit $q = \frac{5}{6}$ braucht man $(1 + E_1)$ Schritte.

Aufgelöst nach E_1 erhält man: $E_1 = \frac{1}{1-q} = \frac{1}{p} = 6$.

Das macht intuitiv Sinn, dass bei jedem sechsten Wurf eine Fünf fallen sollte. Andererseits ist die Intuition auch falsch. Weil wenn die Fünfen regelmässig fallen würden (jeder sechste Wurf wäre eine Fünf), müsste man durchschnittlich nur 3.5 Schritte warten!

f) Es bleiben noch zwei Würfel. Sei $X \sim \text{Bin}(2, \frac{1}{6})$. Damit lässt sich E_2 analog zu Aufgabe e) wie folgt schreiben:

$$\begin{aligned} E_2 &= P(X = 2) \cdot 1 + P(X = 1) \cdot (1 + E_1) + P(X = 0) \cdot (1 + E_2) \\ E_2 &= p^2 \cdot 1 + 2pq \cdot (1 + E_1) + q^2 \cdot (1 + E_2) \\ E_2(1 - q^2) &= p^2 + 2pq + q^2 + 2pq\frac{1}{p} = (p + q)^2 + 2q = 1 + 2q \\ E_2 &= \frac{1 + 2q}{1 - q^2} = \frac{\frac{16}{6}}{\frac{11}{36}} = \frac{96}{11} = 8.\overline{72} \end{aligned}$$

g) Sei $X \sim \text{Bin}(k, \frac{1}{6})$

$$E_k = P(X=k)\cdot 1 + \sum_{i=0}^{k-1} P(X=i)\cdot(1+E_{k-i})$$

$$E_k = P(X=k)\cdot 1 + P(X=0)\cdot(1+E_k) + \sum_{i=1}^{k-1} P(X=i)\cdot(1+E_{k-i})$$

$$E_k\cdot(1-P(X=0)) = P(X=k)\cdot 1 + P(X=0)\cdot 1 + \sum_{i=1}^{k-1} P(X=i)\cdot(1+E_{k-i})$$

$$E_k\cdot(1-P(X=0)) = \sum_{i=0}^{k} P(X=i) + \sum_{i=1}^{k-1} P(X=i)\cdot E_{k-i}$$

$$E_k\cdot(1-P(X=0)) = 1 + \sum_{i=1}^{k-1} P(X=i)\cdot E_{k-i}$$

$$E_k = \frac{1+\sum_{i=1}^{k-1} P(X=i)\cdot E_{k-i}}{1-P(X=0)}$$

h) Die Funktion `binom` nutzt die Identität $\binom{n}{k} = \binom{n}{n-k}$. Berechnet wird die gekürzte Version von $\frac{n!}{k!\cdot(n-k)!} = \frac{n\cdot(n-1)\cdot\ldots\cdot(n-k+1)}{k\cdot(k-1)\cdot\ldots 2\cdot 1}$.

i) Vollständiges Programm

```
import math

# Binomialkoeffizient n tief k
def binom(n,k):
    b = 1
    if k>n/2:
        k=n-k
    for i in range(k):
        b = b * (n-i)/(k-i)
    return b


# Wahrscheinlichkeit, dass eine binomialverteilte Zufallsvariable mit Parametern
# (n,p) den Wert k annimmt.
def binprob(n,k,p):
    return binom(n,k)*p**k*(1-p)**(n-k)



# Liste mit Werten für E_k, start bei E_0 = 0
e = [0]
for k in range(1,200):
    print(k)
    # \frac{1 + \sum_{i=1}^{k-1} P(X=i)\cdot E_{k-i}}{1-P(X=0)}
    s=1
    for i in range(1,k):
        s = s + binprob(k,i,1/6)*e[k-i]
    s = s / (1-binprob(k,0,1/6))
    e.append(s)

# Liste mit Abweichungen unserer Schätzung
d = [e[k] - math.log(k)/math.log(6/5) for k in range(1,len(e))]
print(e)
print(d)
```

j) Die Simulation mit $n = 1'000'000$ Würfel und 100 Wiederholungen hat ergeben:
Schnitt: 78.73, $\sigma_{\overline{x}}$: 0.72528
Mit 500 Wiederholungen wurde erhalten:
Schnitt: 79.16, $\sigma_{\overline{x}}$: 0.31541
Die Anzahl Würfel wird in jedem Schritt (zumindest so lange «viele Würfel» vorhanden sind) mit $\frac{5}{6}$ multipli-

ziert (d.h. $\frac{1}{6}$ der Würfel werden entfernt). Die Anzahl Schritte s sollte ungefähr die Gleichung

$$\begin{aligned} n \cdot \left(\frac{5}{6}\right)^s &\approx 1 \\ \left(\frac{5}{6}\right)^s &\approx \frac{1}{n} \\ s &\approx \log_{\frac{5}{6}}\left(\frac{1}{n}\right) \\ s &\approx \log_{\frac{6}{5}}(n) = \frac{\ln(n)}{\ln(6)-\ln(5)} \end{aligned}$$

Für $n = 1'000'000$ ist $s \approx 75.7$, was nicht im Vertrauensintervall der Simulation liegt, aber auch nicht sehr weit daneben liegt.

k) Die Abweichung $E_k - \frac{\ln(k)}{\ln(6)-\ln(5)}$ scheint sich einem Wert um 3.66 anzuhähern (so weit man Binomialkoeffizienten mit unserem Code noch berechnen kann). Was sich sehr schön mit obigen Versuchen deckt.

Eine Alternative, das Problem anzugehen, besteht darin, einzelne Würfel zu verfolgen, wie lange es geht, bis diese Ausscheiden. Diese Anzahl Schritte ist eine Zufallsvariable die einer soganannten *geometrischen Verteilung* folgt. Von n solchen Zufallsvariablen bestimmt man das Maximum. Eine geschlossene Formel dafür scheint es nicht zu geben. Mehr dazu:

https://de.wikipedia.org/wiki/Geometrische_Verteilung und

https://math.stackexchange.com/questions/26167/