Řešená cvičení: NMAI059 Pravděpodobnost a statistika 1

Karel Král, TODO

10. března 2021

Obsah

Kapitola 1

Zadání

1.1 Cvičení

1. Úvodní informace:
   (a) Slyšíte mě všichni dobře?
   (b) Literatura.
   (c) Pravidla zápočtu (domácí úkoly).

   Řešení: 1
2. Jak se generuje náhoda programem.
   - Python3
   - C++
   - R

   Řešení: 2
3. Připomeňte si definici pravděpodobnostního prostoru (Definice 2.1). Určete, co je
   - *množina elementárních jevů (sample space)*, tedy množina Ω,
   - *prostor jevů (event space)*, tedy množina $\mathcal{F} \subseteq \mathcal{P}(\Omega)$,
   - *pravděpodobnost (probability)*, tedy funkce $\Pr\colon \mathcal{F} \to [0, 1]$

   pro následující příklady:
   (a) Hod spravedlivou alkoholovou trojhrannou tužkou (není to kostka kvůli popisu prostoru jevů):

   ```
   import random
   drink = random.choice([
                   "světlé pivo",
                   "tmavé pivo",
                   "slivovice",
                   ])
   ```
   (b) Uniformně náhodné číslo z intervalu $[0, 1)$.

   ```
   import random
   print(random.random())
   ```

Řešení: 3

4. Dokažte, že $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$.

   Řešení: 4

5. Zopakujte si základní kombinatoriku:

   - Kolik je různých permutací množiny $\{A, B, C\}$?
   - Kolik různých slov skládajících se z písmen $\{A, B\}$ má délku 3?
   - Kolik různých podmnožin množiny $\{A, B, C, D, E\}$ má velikost 3?
   - Kolik různých kombinací s opakováním z množiny $\{A, B, C, D, E\}$ velikosti 3?

   Řešení: 5

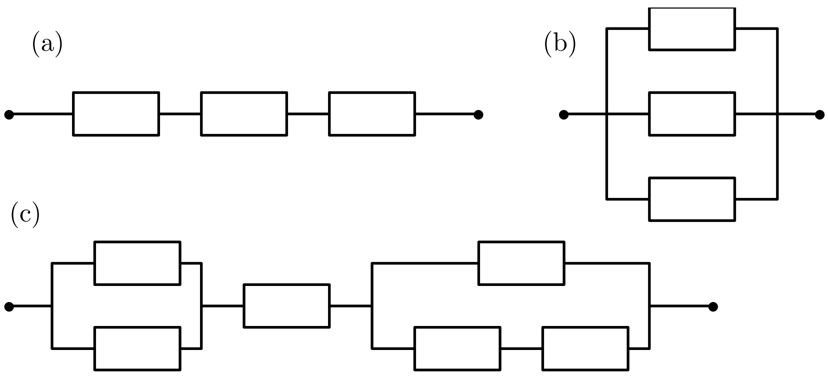
6. Jaká je pravděpodobnost, že při hodu šesti rozlišitelných spravedlivých šestistěnných kostek padnou aspoň na třech kostkách aspoň tři? Jaký je množina elementárních jevů, prostor jevů a pravděpodobnost?

   Řešení: 6

7. Nechť Ω jsou všechny permutace prvních 100 přirozených čísel, prostor jevů jsou všechny podmnožiny Ω a každý elementární jev je stejně pravděpodobný. Označme jev A_j že náhodně zvolená permutace $\pi \in \Omega$ splňuje $\pi(j) = j$ (pro $1 \leq j \leq 100$). Jsou A_1, A_2 nezávislé jevy?

   Řešení: 7

8. Každý obdélník na obrázku je součástka, která se může porouchat s pravděpodobností p. Přesněji řečeno porucha znamená, že skrz ní neteče proud. Poruchy součástek jsou na sobě nezávislé. Jaká je pravděpodobnost, že stále poteče proud mezi dvěma puntíky.



   Řešení: 8

1.2 Cvičení

1. Házíme cinknutou mincí – hlava padne s pravděpodobností $p \in [0, 1)$, orel padne s pravděpodobností $1 - p$. Házíme opakovaně dokud nepadne hlava.

   (a) Jak vypadá pravděpodobnostní prostor?

   (b) Jaká je pravděpodobnost, že hodíme právě třikrát (n-krát)?

   (c) Jaká je pravděpodobnost, že hodíme nejvýš třikrát (n-krát)?

   (d) Jaká je pravděpodobnost, že hodíme lišekrát?

(e) Simulujte předchozí.

Řešení: 1

2. Hodíme cinknutou korunou (panna s pravděpodobností $p_1 \in [0,1]$) a cinknutou dvoukorunou (panna s pravděpodobností $p_2 \in [0,1]$). Oba hody jsou na sobě nezávislé.

   (a) Určete pravděpodobnostní prostor.

   (b) Připomeňte si definici podmíněné pravděpodobnosti (Definice 2.2).

   (c) Spočítejte pravděpodobnost Pr[na obou padne panna | na koruně padne panna].

   (d) Spočítejte pravděpodobnost Pr[na obou padne panna | padne aspoň jedna panna].

   (e) Simulujte:

   Řešení: 2

3. Na louce rostou květiny, které mají buď bílé nebo červené květy. Náhodná květina má bílý květ s pravděpodobností $\Pr[B] = 0.6$, tedy pravděpodobnost že náhodná květina má červený květ je $\Pr[C] = 0.4$. Pravděpodobnost že červená květina je jedovatá je $\Pr[J \mid C] = 0.25$. Pravděpodobnost že bílá květina je jedovatá je $\Pr[J \mid B] = 1/12$. Snědli jsme náhodnou rostlinu a je nám zle, jaká je pravděpodobnost, že ta rostlina měla červený květ?

   Řešení: 3

4. V první krabici je b bílých míčků a c červených míčků, ve druhé krabici také. Napřed vytáhneme jeden míček z první krabice (uniformně náhodně) a dáme ho do druhé krabice. Pak vytáhneme jeden míček z druhé krabice (uniformně náhodně). Jaká je pravděpodobnost, že míček vytažený z druhé krabice je červený?

   (a) Navrhněte vhodný pravděpodobnostní prostor.

   (b) Spočítejte tu pravděpodobnost, kterou jsme chtěli.

   (c) Simulujte.

   Řešení: 4

5. Tento příklad je vymyšlený, zejména čísla nesedí a reálný svět je malinko složitější (tím se ještě budeme zabývat), ale informace v něm nejsou daleko od pravdy. V zemi nám řádí nemoc C.

   - Prostor elementárních jevů (sample space) Ω jsou všichni občané.
   - Označíme $C^+ \subseteq \Omega$ množinu všech lidí, kteří dnes mají aktivní nemoc C, označíme $C^- = \Omega \setminus C^+$ zdravé lidi.
   - Umíme uniformně náhodně samplovat lidi, tedy $\forall \omega \in \Omega\colon \Pr[\{\omega\}] = 1/|\Omega|$ (tady ω je jeden člověk).
   - Test nám pro libovolného člověka odpoví že je člověk zdravý nebo nemocný. Značme $T^+ \subseteq \Omega$ množinu lidí pro které test odpoví, že jsou nemocní. Značme $T^- = \Omega \setminus T^+$ množinu lidí pro které test odpoví, že jsou zdraví. Ale není to tak jednoduché, v příbalovém letáku testu se píše:
     - *Sensitivity:* (true positive) $\Pr[T^+ \mid C^+] = 0.9$
     - *Specificity:* (true negative) $\Pr[T^- \mid C^-] = 0.8$

     z tohoto můžeme odvodit chyby:

     - False positive = false alarm = type I error

$$\Pr[T^+ \mid C^-] = 1 - \Pr[T^- \mid C^-] = 0.2$$

– False negative = miss = type II error

$$\Pr[T^- \mid C^+] = 1 - \Pr[T^+ \mid C^+] = 0.1$$

- Provedli jsme jeden test u každého z uniformně náhodně vybraných 50000 lidí a pozitivních testů vyšlo 1000. Tedy předpokládáme, že $\Pr[T^+] = \frac{1000}{50000} = \frac{1}{50} = 0.02$ (jak moc je tento předpoklad oprávněný budeme zkoumat nadále).
- Zajímá nás $\Pr[C^+|$ (vynásobeno 100 nám dá počet nemocných v procentech).

(a) V čem se toto liší od reality?

(b) Spočítejte $\Pr[C^+]$.

(c) Co se stalo špatně?

(d) Jak by vyšlo předchozí kdyby $\Pr[T^+ \mid C^+] = 0.99, \Pr[T^- \mid C^-] = 0.98, \Pr[T^+] = 0.2$?

(e) Simulujte předchozí.

Řešení: 5

6. V šuplíku mám $b \in \mathbb{N}$ párů bílých, $c \in \mathbb{N}$ párů černých ponožek a $s \in \mathbb{N}$ párů sepraných ponožek. Potřebuju si vytáhnout čtyři páry černých ponožek (jedu na prodloužený víkend tancovat). Když vytáhnu čtyři náhodné páry ponožek (mám je napárované v šuplíku), jaká je pravděpodobnost, že všechny budou černé?

   Řešení: 6

7. Na stole jsou dvě obálky, v jedné je k korun, ve druhé ℓ korun ($k, \ell \in \mathbb{N}$). Můžete otevřít jednu obálku a na základě sumy v ní se rozhodnout jestli si necháte tu otevřenou nebo si vezmete tu druhou (nehledě na to, kolik je v té druhé). Umíte vymyslet způsob jak odejít s tou s větším obnosem s pravděpodobností ostře větší než jedna polovina?

   Řešení: 7

Kapitola 2

Tahák

2.1 Pravděpodobnostní prostor

Definice. Pravděpodobnostní prostor *je trojice* $(\Omega, \mathcal{F}, \Pr)$*, kde*

1. Ω *je* množina elementárních jevů *(sample space) (je to množina)*
2. $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ *je* prostor jevů *(event space) (je to množina podmnožin* Ω*, jednotlivé prvky* $\mathcal{F}$ *nazýváme jevy)* $\mathcal{F}$ *je* σ*-algebra, tedy musí platit:*
   (a) $\emptyset \in \mathcal{F}$ *a zároveň* $\Omega \in \mathcal{F}$ *(celá množina elementárních jevů je jev)*
   (b) $A \in \mathcal{F} \Rightarrow (\Omega \setminus A) \in \mathcal{F}$ *(prostor jevů je uzavřený na doplňky)*
   (c) $(\forall n \in \mathbb{N} \colon A_n \in \mathcal{F}) \Rightarrow (\cup_{n \in \mathbb{N}} A_n) \in \mathcal{F}$ *(prostor jevů je uzavřený na spočetná sjednocení, poznámka může se stát, že* $A_1 \neq A_2 = A_3 = \cdots$*, specielně tedy je uzavřený i na všechna konečná sjednocení)*
3. $\Pr$ *je funkce* $\Pr \colon \mathcal{F} \to [0,1]$ *je* pravděpodobnost *jevu z prostoru jevů, musí splňovat:*
   (a) $\Pr[\emptyset] = 0$ *a zároveň* $\Pr[\Omega] = 1$
   (b) $\Pr$ *je spočetně aditivní, tedy pro každou* $I \subseteq \mathbb{N}$ *a každou posloupnost jevů* $(A_j)_{j \in I}$*, které jsou po dvou disjunktní (tedy* $\forall i, j \in I \colon i \neq j \Rightarrow A_i \neq A_j$*) platí:*

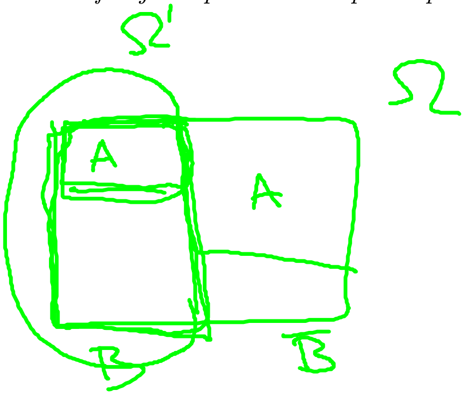
$$\Pr\left[\cup_{j \in I} A_j\right] = \sum_{j \in I} \Pr[A_j]$$

(tedy $\Pr$ *je pravděpodobnostní míra)*

2.2 Podmíněná pravděpodobnost

Definice. *Nechť* $(\Omega, \mathcal{F}, \Pr)$ *je pravděpodobnostní prostor. Nechť* $A, B \in \mathcal{F}$ *a navíc* $\Pr[B] > 0$*. Pak definujeme podmíněnou pravděpodobnost*

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$



2.3 Bayesova věta

Věta 1. *Nechť* $(\Omega, \mathcal{F}, \Pr)$ *je pravděpodobnostní prostor. Nechť* $B_1, B_2, \ldots \in \mathcal{F}$ *je rozklad* Ω *(na spočetně mnoho množin). Nech* $A \in \mathcal{F}$ *a navíc platí* $\Pr[A] > 0$ *a navíc* $\Pr[B_i] > 0$ *pro každé* i*. Pak*

$$\Pr\left[B_j \mid A\right] = \frac{\Pr[A \mid B_j] \Pr[B_j]}{\sum_i \Pr[A \mid B_i] \Pr[B_i]}$$

2.4 Nezávislé jevy

Definice. *Nechť* $(\Omega, \mathcal{F}, \Pr)$ *je pravděpodobnostní prostor. Nechť* $A, B \in \mathcal{F}$ *jsou dva jevy. Pak řekneme, že* A, B *jsou* nezávislé jevy*, pokud platí:*

$$\Pr[A \cap B] = \Pr[A] \Pr[B]$$

(také se dá říct $\Pr[A \mid B] = \Pr[A]$ *pokud* $\Pr[B] > 0$*).*

Kapitola 3

Řešení

3.1 Cvičení

1. **Úvodní informace:**
   (a) **Slyšíte mě všichni dobře?**
   (b) **Literatura.**
       Řešení: TODO
   (c) **Pravidla zápočtu (domácí úkoly).**
       Řešení: TODO

2. **Jak se generuje náhoda programem.**

- **Python3** ***Řešení:***

```python
# https://docs.python.org/3/library/random.html
# Nepoužívejte pro šifrování!
import random
import scipy

# https://docs.python.org/3/library/itertools.html
import itertools

# Generuj náhodné celé číslo 1 <= x <= 10 (tedy x in range(1, 11))
x = random.randint(1, 10)
print(x)

# Generuj náhodný prvek dané neprázdné posloupnosti.
drink = random.choice([
    "světlé pivo",
    "tmavé pivo",
    "slivovice",
    "bílé víno",
    "červené víno",
    "čaj",
    ])
print(drink)

# Výsledek 'B' je třikrát pravděpodobnější než 'A'.
print(random.choice(['A', 'B', 'B', 'B']))

# Vrátí list k náhodných prvků z dané sekvence s danými váhami
# (cum_weights jsou trochu rychlejší).
# https://docs.python.org/3/library/random.html#random.choices
print(random.choices(['A', 'B', 'C'], weights=[1/6, 1/6, 2/3], k=5))

# Náhodná permutace (mění přímo daný list).
my_list = ['a', 'b', 'c', 'd']
random.shuffle(my_list)
print(my_list)

# Vybere dva prvky dané posloupnosti, ve výsledku se nebudou opakovat
# pozice. Pokud se prvky opakují, tak se mohou ve výsledku opakovat.
print(random.sample(['w', 'x', 'y', 'z'], k=2))  # two distinct letters
print(random.sample(['w', 'x', 'y', 'x'], k=2))  # 'x' can repeat

# Pro přesné počítání (vrací iterable):
print(list(itertools.permutations([1, 2, 3])))
print(list(itertools.combinations('ABCD', 2)))
print(list(itertools.combinations_with_replacement('ABCD', 2)))

# Může se hodit scipy: sudo apt-get install python3-scipy
scipy.special.comb(n=5, k=2, exact=True)  # vrátí n nad k
```

- **C++** ***Řešení:***

```cpp
std::default_random_engine generator;
```

```
std::uniform_int_distribution<int> distribution(0, 9);
```

- **R** ***Řešení:***

```
x <- "hello"
```

3. **Připomeňte si definici pravděpodobnostního prostoru (Definice 2.1). Určete, co je**

   - ***množina elementárních jevů (sample space)*, tedy množina Ω,**
   - ***prostor jevů (event space)*, tedy množina $\mathcal{F} \subseteq \mathcal{P}(\Omega)$,**
   - ***pravděpodobnost (probability)*, tedy funkce $\Pr\colon \mathcal{F} \to [0,1]$**

   pro následující příklady:

   (a) **Hod spravedlivou alkoholovou trojhrannou tužkou (není to kostka kvůli popisu prostoru jevů):**

   ```
   import random
   drink = random.choice([
                   "světlé pivo",
                   "tmavé pivo",
                   "slivovice",
                   ])
   ```

   Řešení: V počítači nám stačí předchozí kód (pseudonáhodné číslo vs náhodné číslo). Fyzicky můžeme generovat pomocí hodu trojhrannou tužkou, která má značky na stěnách (a zaručeně nemůže padnout nastojato).

   - Množina elementárních jevů $\Omega = \{\text{světlé pivo}, \text{tmavé pivo}, \text{slivovice}\}$
   - Prostor jevů $\mathcal{F} = \mathcal{P}(\Omega)$, tedy $|\mathcal{F}| = 2^{|\Omega|} = 2^3$:

   $$\begin{aligned}\mathcal{F} = \{&\\ &\emptyset,\\ &\{\text{světlé pivo}\},\\ &\{\text{tmavé pivo}\},\\ &\{\text{slivovice}\},\\ &\{\text{světlé pivo}, \text{tmavé pivo}\},\\ &\{\text{světlé pivo}, \text{slivovice}\},\\ &\{\text{tmavé pivo}, \text{slivovice}\},\\ &\{\text{světlé pivo}, \text{tmavé pivo}, \text{slivovice}\}\\ \}&\end{aligned}$$

   - Pravděpodobnost

   $$\begin{aligned}\Pr[\emptyset] &= 0\\ \Pr[\{\text{světlé pivo}\}] &= 1/3\\ \Pr[\{\text{tmavé pivo}\}] &= 1/3\\ \Pr[\{\text{slivovice}\}] &= 1/3\\ \Pr[\{\text{světlé pivo}, \text{tmavé pivo}\}] &= 2/3\\ \Pr[\{\text{světlé pivo}, \text{slivovice}\}] &= 2/3\\ \Pr[\{\text{tmavé pivo}, \text{slivovice}\}] &= 2/3\\ \Pr[\{\text{světlé pivo}, \text{tmavé pivo}, \text{slivovice}\}] &= 1\end{aligned}$$

   Většinou ale nepopisujeme jev jako podmnožinu elementárních jevů, ale nějak lidsky:

   $$\Pr[\text{nějaké pivo}] = \Pr[\text{světlé pivo} \vee \text{tmavé pivo}] = \Pr[\{\text{světlé pivo}, \text{tmavé pivo}\}] = 2/3$$

$$\Pr[\text{ne pivo}] = \Pr[\{\text{slivovice}\}] = 1 - \Pr[\{\text{světlé pivo, tmavé pivo}\}] = 1/3$$

(b) Uniformně náhodné číslo z intervalu $[0, 1)$**.**

```
import random
print(random.random())
```

Řešení: V počítači nám funkce `random.random` vrátí float x, který je přesně reprezentovatelný, platí $0.0 \leq x < 1.0$ a zároveň x je celočíselný násobek 2^{-53}. To ale znamená, že nikdy nedostaneme 0.05954861408025609 i když to je číslo přesně reprezentovatelné jako python float. Můžeme generovat i vícebitová čísla, ale vždy to bude číslo! A to je dobře, většina reálných čísel nejde reprezentovat konečnou posloupností symbolů (pozor, $\sqrt{2}$ jde reprezentovat jako kořen polynomu, ale i třeba $1/\pi$ jde reprezentovat například algoritmem, který ho počítá).

Jak bychom fyzikálně generovali uniformně náhodné číslo z intervalu $[0, 1)$? Uniformně náhodné znamená, že každé číslo bude stejně pravděpodobné. Hod šipkou na interval má nevýhodu, že buď budou konce intervalu méně pravděpodobné než prostředek nebo se nám může stát že hodíme šipku mimo interval (nejspíš obojí). Můžeme ale udělat terč s jedním vyznačeným poloměrem, který bude kruh, připevnit jeho střed k vrtačce, roztočit a hodit šipku (tak abychom netrefili střed, ale určitě trefili kruh). Pak náhodné číslo $x \in [0, 1)$ bude úhel který svírá vyznačný poloměr a naše šipka dělený 2π.

- Toto je jen myšlenkový experiment, fyzická implementace je nejspíš poměrně nebezpečná. Doma to nezkoušejte!
- Můžete namítnout, že šipka nevybere přesně bod, že terč je tvořen atomy a tedy je také z nějakého pohledu diskrétní. Asi ano, ale já neříkal, že reálná čísla existují. Pro představu atom může mít poloměr okolo $10^{-10}m$, tedy na délce $1m$ jich vedle sebe vyskládáme zhruba 10^{10}. Srovnejte s přesností $2^{-53} \approx 10^{-16}$, tedy pokud bychom výsledek `random.random()` brali jako pozici v metrech, pak máme přesnost zhruba na dvě miliontiny atomu.
- *množina elementárních jevů (sample space)*, tedy množina $\Omega = [0, 1)$
- *prostor jevů (event space)*, tedy množina $\mathcal{F} \subseteq \mathcal{P}(\Omega)$

  Napřed se zeptejme, jestli by nemohlo platit, že $\mathcal{F} = \mathcal{P}(\Omega)$. Asi bychom chtěli následující vlastnosti:

  – pokud $A \in \mathcal{F}$ je interval, pak $\Pr[A]$ je rovna délce A

  – pokud $A \in \mathcal{F}$ a navíc $x + A = \{x + a \mid a \in A\} \subseteq \Omega$ pro nějaké reálné číslo x, pak $\Pr[A] = \Pr[x + A]$.

  – každá podmnožina Ω má přiřazenou nějakou pravděpodobnost

  Ale to nejde, protože ne každá množina má míru (tady Pr odpovídá takzvané pravděpodobnostní míře – míra celého prostoru je rovna jedné). Nejznámějším příkladem je `https://en.wikipedia.org/wiki/Banach%E2%80%93Tarski_paradox`. Hezké video: `https://www.youtube.com/watch?v=s86-Z-CbaHA`.

  Naše řešení: $\mathcal{F}$ je množina Lebesgueovsky měřitelných podmnožin Ω.
- *pravděpodobnost (probability)*, tedy funkce $\Pr\colon \mathcal{F} \to [0, 1]$ je Lebesgueova míra.

Pokud jste neslyšeli o tom, co je to míra, tak se nelekejte. Dobré k zapamatování:

- Pravděpodobnost je číslo mezi nulou a jedničkou (obecná míra celého prostoru nemusí být rovna jedné, ale nás zajímá pravděpodobnostní míra).

- "Nic se nestane" má pravděpodobnost nula – $\Pr[\emptyset] = 0$ (tedy specielně $\emptyset$ je měřitelná).
- "Něco se stane" má pravděpodobnost jedna – $\Pr[\Omega] = 1$ (tedy specielně Ω je měřitelná).
- "Stane se A" má pravděpodobnost 1-"Nestane se A" – $\Pr[A] = 1 - \Pr[\Omega \setminus A]$ (tedy pokud je A měřitelná, pak je i její doplněk měřitelný).
- Pr[Na kostce padne jedna tečka nebo dvě tečky] = Pr[padne jedna tečka]+Pr[padnou dvě tečky] – pravděpodobnost sjednocení disjunktních množin je rovna součtu jejich pravděpodobností (platí také pro *spočetně* mnoho disjunktních množin a navíc sjednocení spočetně mnoha disjunktních měřitelných množin je taky měřitelné)
- Lebesgueova míra jednoho bodu je nulová (tedy ze spočetného disjunktního sjednocení máme že pravděpodobnost že uniformně náhodné reálné číslo z intervalu $[0, 1)$ je racionální je nulová).

  Pozor na to, že sice platí $\Pr[\{0.1\}] = 0$, ale to neznamená, že jev že padne 0.1 je nemožný (akorát velice velice nepravděpodobný). Naopak pokud je jev nemožný Pr[na šestistěnné kostce padne sedm], pak je jeho pravděpodobnost nulová.
- Úsečka v $[0, 1] \times [0, 1] \subseteq \mathbb{R}^2$ má Lebesgueovu míru nula.

Takže ty definice, které vám přijdou divné jsou tam kvůli teorii míry (případně později kvůli teorii integrálu).

Pozor na to, že ne každá pravděpodobnost je Lebesgueova míra, můžeme uvažovat například $\Omega = [0, 1)$, $\mathcal{F}$ jsou Lebesgueovsky měřitelné, Pr která $\Pr[\{0.1\}] = 1/2$ a $\Pr[A] = \lambda(A)/2 + 1/2$ pokud $0.1 \in A$ a $\Pr[A] = \lambda(A)/2$ jinak (kde $\lambda(A)$ značí Lebesgueovu míru množiny A).

Někdy také mluvíme o pravděpodobnostní distribuci, zatím to můžete brát jako synonymum k pravděpodobnosti.

4. **Dokažte, že** $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$**.**

Řešení: Předpokládejme, že platí $A \cap B, A \cup B, A, B \in \mathcal{F}$ (jinak by výraz nahoře neměl smysl). Z definice pravděpodobnostního prostoru (Definice 2.1) máme spočetnou aditivitu pro disjunktní jevy. Rozdělíme tedy $A \cup B$ na disjunktní množiny:

$$\begin{aligned} C_1 &= A \cap B \\ C_2 &= A \setminus B \\ C_3 &= B \setminus A \end{aligned}$$

tedy máme $A \cup B = C_1 \cup C_2 \cup C_3$ a zároveň $C_i \cap C_j = \emptyset$ pro každé dvě $1 \leq i < j \leq 3$. Dle spočetné aditivity můžeme psát:

$$\begin{aligned} \Pr[A \cup B] &= \Pr[C_1 \cup C_2 \cup C_3] \\ &= \Pr[C_1] + \Pr[C_2] + \Pr[C_3] \end{aligned}$$

Máme také:

$$\begin{aligned} A &= C_2 \cup C_1 = (A \setminus B) \cup (A \cap B) \\ B &= C_3 \cup C_1 = (B \setminus A) \cup (A \cap B) \end{aligned}$$

takže můžeme psát:

$$\begin{aligned} \Pr[A \cup B] &= \Pr[C_1] + \Pr[C_2] + \Pr[C_3] \\ &= \Pr[C_1] + \Pr[C_2] + 2\Pr[C_3] - \Pr[C_3] \\ &= (\Pr[C_1] + \Pr[C_3]) + (\Pr[C_2] + \Pr[C_3]) - \Pr[C_3] \\ &= (\Pr[A]) + (\Pr[B]) - \Pr[C_3] \\ &= \Pr[A] + \Pr[B] - \Pr[A \cap B] \end{aligned}$$

což jsme chtěli dokázat.

5. **Zopakujte si základní kombinatoriku:**

- **Kolik je různých permutací množiny $\{A, B, C\}$?**

  Řešení: $3! = 3 \cdot 2 \cdot 1$, obecně $n!$ pokud máme n rozlišitelných prvků

```
import itertools

permutations = list(itertools.permutations(['A', 'B', 'C']))
print(f'There are {len(permutations)} permutations: {permutations}')

# There are 6 permutations:
#    [('A', 'B', 'C'),
#     ('A', 'C', 'B'),
#     ('B', 'A', 'C'),
#     ('B', 'C', 'A'),
#     ('C', 'A', 'B'),
#     ('C', 'B', 'A')]
```

- **Kolik různých slov skládajících se z písmen $\{A, B\}$ má délku 3?**

  Řešení: $2^3 = 8$, obecně n^r kde n je počet písmen, r délka slova

```
import itertools

all_words = list(itertools.product('AB', repeat=3))
print(f'There are {len(all_words)} words: {all_words}')

# There are 8 words:
#    [('A', 'A', 'A'),
#     ('A', 'A', 'B'),
#     ('A', 'B', 'A'),
#     ('A', 'B', 'B'),
#     ('B', 'A', 'A'),
#     ('B', 'A', 'B'),
#     ('B', 'B', 'A'),
#     ('B', 'B', 'B')]
```

- **Kolik různých podmnožin množiny $\{A, B, C, D, E\}$ má velikost 3?**

  Řešení: $\binom{n}{r} = \frac{n!}{r!(n-r)!} = \binom{5}{3} = 10$

```
import itertools

all_subsets = list(itertools.combinations('ABCDE', 3))
print(f'There are {len(all_subsets)} subsets: {all_subsets}')

# There are 10 subsets:
#     [('A', 'B', 'C'),
#      ('A', 'B', 'D'),
#      ('A', 'B', 'E'),
#      ('A', 'C', 'D'),
#      ('A', 'C', 'E'),
#      ('A', 'D', 'E'),
#      ('B', 'C', 'D'),
#      ('B', 'C', 'E'),
#      ('B', 'D', 'E'),
#      ('C', 'D', 'E')]
```

- **Kolik různých kombinací s opakováním z množiny $\{A, B, C, D, E\}$ velikosti 3?**

  Řešení: $\binom{n+r-1}{r}$ kde $n = 5$, $r = 3$. Protože máme $n-1$ svislítek a r hvězdiček a kódujeme takto:

  $$AAD = **|||*|$$

  tedy

  $$\text{počet A}|\text{počet B}|\text{počet C}|\text{počet D}|\text{počet E}$$

  kde každý počet je reprezentován počtem hvězdiček a vybíráme které z $n+r-1$ symbolů budou hvězdičky.

```python
import itertools

sorted_sequences = list(itertools.combinations_with_replacement('ABCDE', r=3))
print(f'Máme {len(sorted_sequences)} sekvencí: {sorted_sequences}')

# There are 35 sorted sequences:
#    [('A', 'A', 'A'), ('A', 'A', 'B'), ('A', 'A', 'C'),
#     ('A', 'A', 'D'), ('A', 'A', 'E'), ('A', 'B', 'B'),
#     ('A', 'B', 'C'), ('A', 'B', 'D'), ('A', 'B', 'E'),
#     ('A', 'C', 'C'), ('A', 'C', 'D'), ('A', 'C', 'E'),
#     ('A', 'D', 'D'), ('A', 'D', 'E'), ('A', 'E', 'E'),
#     ('B', 'B', 'B'), ('B', 'B', 'C'), ('B', 'B', 'D'),
#     ('B', 'B', 'E'), ('B', 'C', 'C'), ('B', 'C', 'D'),
#     ('B', 'C', 'E'), ('B', 'D', 'D'), ('B', 'D', 'E'),
#     ('B', 'E', 'E'), ('C', 'C', 'C'), ('C', 'C', 'D'),
#     ('C', 'C', 'E'), ('C', 'D', 'D'), ('C', 'D', 'E'),
#     ('C', 'E', 'E'), ('D', 'D', 'D'), ('D', 'D', 'E'),
#     ('D', 'E', 'E'), ('E', 'E', 'E')]
```

6. **Jaká je pravděpodobnost, že při hodu šesti rozlišitelných spravedlivých šestistěnných kostek padnou aspoň na třech kostkách aspoň tři? Jaký je množina elementárních jevů, prostor jevů a pravděpodobnost?**

Řešení:

- Množina elementárních jevů je $\Omega = \{1, 2, 3, 4, 5, 6\}^6 = \{111111, 111112, \ldots, 666666\}$, tedy $|\Omega| = 6^6 = 46656$.
- Prostor jevů je $\mathcal{F} = \mathcal{P}(\Omega)$, tedy $|\mathcal{F}| = 2^{46656}$
- Pravděpodobnost $\Pr[\{abcdef\}] = 1/6^6$ pro libovolná $a, b, c, d, e, f \in \{1, 2, 3, 4, 5, 6\}$.

Jak takový příklad řešit? Uvědomit si přesně o čem mluvíme, pak zkusit přemýšlet o jednodušších jevech.

- Na jedné kostce padnou aspoň tři s pravděpodobností $4/6 = 2/3$ (musí padnout $3, 4, 5, 6$, tedy nepadne $1, 2$).
- Pokud vybereme k kostek, pak pravděpodobnost, že přesně na těchto kostkách padne aspoň tři je přesně $(2/3)^k(1/3)^{6-k}$ (kostky jsou nezávislé). Kupříkladu pokud vybereme první čtyři kostky, pak nás zajímá:

$$\Pr\left[\{ABCDef \mid A, B, C, D \in \{3, 4, 5, 6\}, e, f \in \{1, 2\}\}\right] = \frac{4^4 \cdot 2^2}{6^6} = (2/3)^4(1/3)^{6-4}$$

- Přesně k kostek vybereme $\binom{6}{k}$ způsoby.
- Rozdělíme pravděpodobnostní prostor na jevy, kde přesně na k kostkách padne číslo aspoň tři (tedy máme disjunktní rozklad) a $k \geq 3$, tedy použijeme definici pravděpodobnosti a sečteme předchozí pro $k \in \{3, 4, 5, 6\}$:

$$\begin{aligned}\Pr[\text{aspoň na třech kostkách aspoň tři}] =& \binom{6}{3}(2/3)^3(1/3)^{6-3} \\ &+ \binom{6}{4}(2/3)^4(1/3)^{6-4} \\ &+ \binom{6}{5}(2/3)^5(1/3)^{6-5} \\ &+ \binom{6}{6}(2/3)^6(1/3)^{6-6} \\ =& 0.8998628257887514\end{aligned}$$

  Tady jsme vlastně použili větu z přednášky, že pokud $B_0, B_1, \ldots, B_{2^6-1}$ jsou rozklad Ω (tedy $B_i \neq B_j$ pro $i \neq j$ a zároveň $\cup B_i = \Omega$), pak $\Pr[A] = \sum_i \Pr[A \mid B_i] \Pr[B_i]$. Kde jev A je že na aspoň třech kostkách padne aspoň tři. Jev B_i je že na přesně určených kostkách padne aspoň tři (tedy jevy B_i, B_j jsou opravdu disjunktní). Konkrétně 22 zapsané binárně je 010110, pak jev B_{22} je jev že na druhé, čtvrté a páté kostce padlo číslo aspoň tři. Pak $\Pr[A \mid B_x] = 1$ pokud x má v binárním zápisu aspoň tři jedničky a $\Pr[A \mid B_x] = 0$ jinak. Už jsme spočítali, že $\Pr[B_x] = (2/3)^k(1/3)^{6-k}$ pokud x má v binárním zápisu k jedniček.

Pomocí programu:

```python
import itertools
import scipy.special
import random
```

```
# Přesný výsledek pomocí kombinatoriky:

def p(k):
    """ Probability that there are exactly k out of 6 dice with at least 3. """
    return scipy.special.comb(6, k, exact=True) * ((2/3)**k) * ((1/3)**(6-k))


exact_computed = sum(p(k) for k in range(3, 7))
print(f'Přesný výsledek: {exact_computed}')


# Přesný výsledek spočítaný hrubou silou:

def indicator(dice):
    """ Return 1 if at least three dice have at least 3, otherwise
    return 0. """
    if sum(1 for x in dice if x >= 3) >= 3:
        return 1
    else:
        return 0


all_outcomes = itertools.product(range(1, 7), repeat=6)
exact_bruteforce = sum(indicator(d) for d in all_outcomes) / (6**6)
print(f'Hrubá síla: {exact_bruteforce}')


# Simulace:

N = 1000  # Number of tries
simulated = sum(indicator(random.choices(range(1, 7), k=6))
                for _ in range(N)) / N
print(f'Simulace: {simulated}')

# Možný výsledek:
# Přesný výsledek: 0.8998628257887514
# Hrubá síla: 0.8998628257887518
# Simulace: 0.903
```

Porovnejme naše metody:

- Výpočet vzorcem:
  - přesný výsledek
  - potřebovali jsme kombinatoriku a přemýšlet
  - pokud se změní zadání, tak řešení se změní celkem dost
  - velice rychlý výpočet
- Procházení všech možností:
  - jednodušší vymýšlení
  - potřebujeme programovat

  - přesný výsledek (liší se v posledních místech floatu, dáno nepřesnostmi floatové reprezentace, `exact=True` nevrací nativní float)
  - pokud se změní zadání, tak se řešení skoro nezmění
  - pokud je množina elementárních jevů velká, tak se tento postup nepoužitelný

- Simulace:
  - jednoduché vymýšlení (skoro jako předchozí případ)
  - pokud se změní zadání, tak se řešení skoro nezmění
  - časová složitost není lineární ve velikosti množiny elementárních jevů (N krát vybíráme náhodný prvek Ω, což často zvládáme v $\mathcal{O}(\log |\Omega|)$ krocích)
  - nepřesný výsledek – závisí na náhodných bitech počítače a počtu pokusů
  - budeme potřebovat trochu teorie abychom odhadli jak jistí si jsme výsledkem

7. **Nechť Ω jsou všechny permutace prvních 100 přirozených čísel, prostor jevů jsou všechny podmnožiny Ω a každý elementární jev je stejně pravděpodobný. Označme jev A_j že náhodně zvolená permutace $\pi \in \Omega$ splňuje $\pi(j) = j$ (pro $1 \le j \le 100$). Jsou A_1, A_2 nezávislé jevy?**

   Řešení:

   - Počet permutací v A_j je přesně 99! (jeden prvek je fixní, zbytek permutujeme), tedy
   $$\Pr[A_j] = \frac{99!}{100!} = \frac{1}{100}$$
   - Počet permutací v $A_1 \cap A_2$ je přesně 98! (dva prvky jsou fixní, zbytek permutujeme), tedy
   $$\Pr[A_1 \cap A_2] = \frac{98!}{100!} = \frac{1}{9900}$$
   - Dle definice nezávislých jevů bychom potřebovali $\Pr[A_1]\Pr[A_2] = \Pr[A_1 \cap A_2]$ (Definice 2.4), ale to neplatí:
   $$\Pr[A_1 \cap A_2] = \frac{1}{9900} \neq \frac{1}{10000} = \Pr[A_1]\Pr[A_2]$$

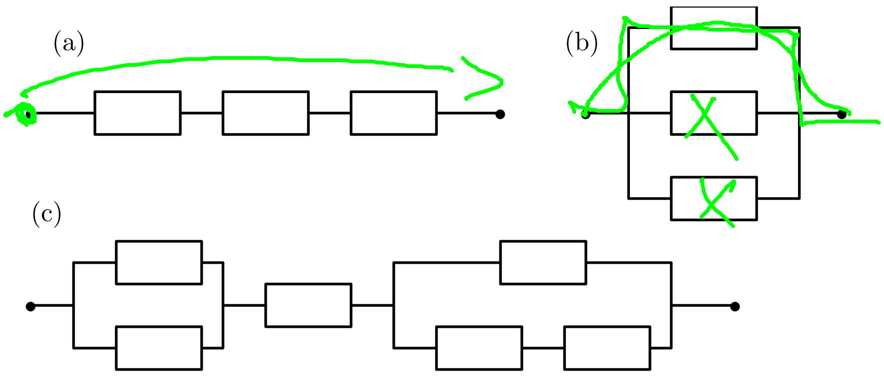
   Zamysleme se nad počítačovým řešením:

```python
import random


# indexujeme od nuly
def fixed(my_list, j):
    return my_list[j] == j


my_list = list(range(100))
N = 100000

A1 = 0
for _ in range(N):
    random.shuffle(my_list)
    A1 += 1 if fixed(my_list, 0) else 0
A1 = A1 / N
print(f'Pr[A_1] = Pr[A_2] = {A1} (= {1/100})')

A1A2 = 0
for _ in range(N):
    random.shuffle(my_list)
    A1A2 += 1 if fixed(my_list, 0) and fixed(my_list, 1) else 0
A1A2 = A1A2 / N
print(f'Pr[A_1 and A_2] = {A1A2} (= {1/9900})')

# Možný výsledek:
# Pr[A_1] = Pr[A_2] = 0.00993 (= 0.01)
# Pr[A_1 and A_2] = 0.00012 (= 0.00010101010101010101)
```

   - jednoduchá simulace

- potřebujeme více pokusů, protože potřebujeme odhadnout s větši přesností (menší pravděpodobnost, tak abychom nedostali nulu)
- vůbec nemůžeme použít hrubou sílu, neboť $100! \approx 9.33 \cdot 10^{157}$, pro představu:
  - počítač vykoná zhruba 10^9 instrukcí za sekundu
  - lineární algoritmus (další permutaci najdeme v jednotkovém čase) by trval zhruba 10^{148} sekund
  - stáří vesmíru se odhaduje na $13.787 \cdot 10^9$ let
  - jeden rok trvá zhruba $\pi \cdot 10^7$ sekund
  - stáří vesmíru je tedy zhruba $4.34 \cdot 10^{17}$ sekund
  - tedy lineární algoritmus který by prošel všechny permutace 100 prvkové množiny by běžel zhruba 10^{131} stáří vesmíru

8. **Každý obdélník na obrázku je součástka, která se může porouchat s pravděpodobností p. Přesněji řečeno porucha znamená, že skrz ní neteče proud. Poruchy součástek jsou na sobě nezávislé. Jaká je pravděpodobnost, že stále poteče proud mezi dvěma puntíky.**

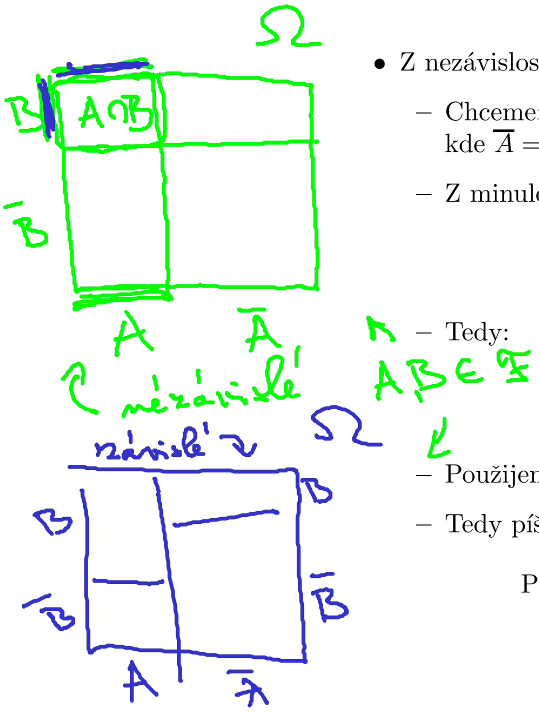


Řešení:

(a) Jak postupujeme:

- Jedna součástka se neporouchá (tedy je ok, jev O, porouchá jev P) s pravděpodobností $\Pr[O] = 1 - \Pr[P] = 1 - p$.
- Aby proud tekl, tak všechny součástky musí být ok. Tedy nás zajímá

$$\Pr[O_1 \cap O_2 \cap O_3]$$



- Z nezávislosti jevů P_1, P_2 máme i nezávislost jevů O_1, O_2 (tedy jejich doplňků):
  - Chceme: $\Pr[A \cap B] = \Pr[A]\Pr[B]$ právě tehdy když $\Pr[\overline{A} \cap \overline{B}] = \Pr[\overline{A}]\Pr[\overline{B}]$ kde $\overline{A} = \Omega \setminus A$ je doplněk
  - Z minulého příkladu přeuspořádáním dostaneme (pro libovolné jevy)

$$\Pr[A \cap B] = \Pr[A] + \Pr[B] - \Pr[A \cup B]$$

  - Tedy:

$$\Pr[\overline{A} \cap \overline{B}] = \Pr[\overline{A}] + \Pr[\overline{B}] - \Pr[\overline{A} \cup \overline{B}]$$

  - Použijeme že $\overline{A} \cup \overline{B} = \overline{A \cap B}$
  - Tedy píšeme:

$$\begin{aligned}
\Pr[\overline{A} \cap \overline{B}] &= \Pr[\overline{A}] + \Pr[\overline{B}] - \Pr[\overline{A} \cup \overline{B}] \\
&= \Pr[\overline{A}] + \Pr[\overline{B}] - \Pr[\overline{A \cap B}] \\
&= (1 - \Pr[A]) + (1 - \Pr[B]) - (1 - \Pr[A \cap B]) \\
&= 1 - \Pr[A] - \Pr[B] + \Pr[A]\Pr[B] \qquad \text{(z nezávislosti } A, B)
\end{aligned}$$

  - Chtěli jsme $\Pr[\overline{A} \cap \overline{B}] = \Pr[\overline{A}]\Pr[\overline{B}] = (1 - \Pr[A])(1 - \Pr[B])$, což je akorát jinak napsaný předchozí řádek.

- Z nezávislosti jevů O_1, O_2, O_3 máme rovnou

$$\begin{aligned}\Pr[O_1 \cap O_2 \cap O_3] &= \Pr[O_1]\Pr[O_2]\Pr[O_3] \\ &= (1 - \Pr[P_1])(1 - \Pr[P_2])(1 - \Pr[P_3]) \\ &= (1-p)^3\end{aligned}$$

(b) Druhý příklad je podobný, ale potřebujeme aby aspoň jedna součástka fungovala, tedy chceme

$$\Pr[O_1 \cup O_2 \cup O_3]$$

Jako nápovědu použijte pozorování že $\Pr[A \cup B] = \Pr[A] + \Pr[B] - \Pr[A \cap B]$.

Jednodušší postup je, uvědomit si, že se nemůžou porouchat všechny a použít nezávislost, tedy

$$\begin{aligned}\Pr[O_1 \cup O_2 \cup O_3] &= 1 - \Pr[P_1 \cap P_2 \cap P_3] \\ &= 1 - \Pr[P_1]\Pr[P_2]\Pr[P_3] \\ &= 1 - p^3\end{aligned}$$

(c) Kombinace myšlenek předchozích.

Samozřejmně můžeme simulovat (všimněte si, jak se podmínka v indikátoru mechanicky překlápí na vzorec pravděpodobnosti):

```
from random import choices
from typing import Sequence

# Pravděpodobnost chyby konkrétní součástky (chyby jsou nezávislé).
p = 0.1
# Počet pokusů.
N = 1000000


def bernoulli_list(k: int, pr: float = p) -> Sequence[bool]:
    """Vrací k samplů z Bernoulliho rozdělení s pravděpodobností pr."""
    return choices([True, False], weights=[pr, 1-pr], k=k)
```

=3

[F , F , F]

```
# a)
#     +------+   +------+   +------+
# o---|   0  |---|   1  |---|   2  |---o
#     +------+   +------+   +------+

def proud_tece_a(soucastky: Sequence[bool]) -> int:
    """Indikátor, jestli proud teče.
    soucastky[i] = i-tá součástka je porouchaná."""
    assert len(soucastky) == 3
    return int(not soucastky[0] and not soucastky[1] and not soucastky[2])
```

∧ ∧

```
# Kolikrát proud tekl z N pokusů.
proud_tekl_a = sum(proud_tece_a(bernoulli_list(3)) for _ in range(N))
```

(1-p) · (1-p) · (1-p)


```python
simulation_a = proud_tekl_a / N
answer_a = (1 - p)**3
error_a = abs(answer_a - simulation_a)
print(f'a) Simulováno: {simulation_a} (chyba {error_a})')


# b)
#         +------+
#     +---|  0   |---+
#     |   +------+   |
#     |              |
#     |   +------+   |
# o---+---|  1   |---+---o
#     |   +------+   |
#     |              |
#     |   +------+   |
#     +---|  2   |---+
#         +------+

def proud_tece_b(soucastky: Sequence[bool]) -> int:
    """Indikátor, jestli proud teče.
    soucastky[i] = i-tá součástka je porouchaná."""
    assert len(soucastky) == 3
    return int(not (soucastky[0] and soucastky[1] and soucastky[2]))


proud_tekl_b = sum(proud_tece_b(bernoulli_list(3)) for _ in range(N))

simulation_b = proud_tekl_b / N
answer_b = (p**3)
error_b = abs(answer_b - simulation_b)
print(f'b) Simulováno: {simulation_b} (chyba {error_b})')


# c)
#         +------+                            +------+
#     +---|  0   |---+              +---------|  3   |---------+
#     |   +------+   |   +------+   |         +------+         |
# o---+              +---|  2   |---+                          +---o
#     |   +------+   |   +------+   |   +------+   +------+    |
#     +---|  1   |---+              +---|  4   |---|  5   |----+
#         +------+                      +------+   +------+

def proud_tece_c(soucastky: Sequence[bool]) -> int:
    """Indikátor, jestli proud teče.
    soucastky[i] = i-tá součástka je porouchaná."""
    assert len(soucastky) == 6
    return int(not (soucastky[0] and soucastky[1])
               and not soucastky[2]
               and (not (soucastky[3] and (soucastky[4] or soucastky[5]))))


proud_tekl_c = sum(proud_tece_c(bernoulli_list(6)) for _ in range(N))
```

```
simulation_c = proud_tekl_c / N
answer_c = (1 - (p**2)) * (1 - p) * (1 - (p * (1 - ((1 - p)**2))))
error_c = abs(answer_c - simulation_c)
print(f'c) Simulováno: {simulation_c} (chyba {error_c})')

# Možný výsledek:
# a) Simulováno: 0.728696 (chyba 0.00030400000000000082)
# b) Simulováno: 0.999005 (chyba 5.000000000032756e-06)
# c) Simulováno: 0.874083 (chyba 1.2000000000012001e-05)
```

$\Omega = \{H, OH, OOH, \ldots\}$

3.2 Cvičení

1. **Házíme cinknutou mincí – hlava padne s pravděpodobností $p \in [0, 1)$, orel padne s pravděpodobností $1 - p$. Házíme opakovaně dokud nepadne hlava.**

   (a) **Jak vypadá pravděpodobnostní prostor?**

   Řešení:

   - *Množina elementárních jevů:* Série hodů, které uvažujeme můžeme kódovat pomocí H pokud padne hlava, O pokud padne orel, takže bychom mohli možné série hodů reprezentovat jako

     $$\{H, OH, OOH, OOOH, OOOOH, \ldots\}.$$

     Ale to je poněkud nepraktické, raději budeme reprezentovat počtem hodů (poslední je určitě hlava, ty před ním jsou orlové):

     $$\Omega = \{1, 2, 3, 4, \ldots\}$$

   - *Prostor jevů:* Máme spočetnou množinu elementárních jevů, takže můžeme brát jako prostor jevů celou potenční množinu množiny elementárních jevů: $\mathcal{F} = \mathcal{P}(\Omega)$.

     Lehké cvičení z matematické analýzy: dokažte, že pokud každému elementárnímu jevu přiřadíme pravděpodobnost, tedy

     $$\boxed{\forall \omega \in \Omega\colon \ \Pr[\{\omega\}] \in [0, 1]}$$

     pak víme, že

     –
     $$\Pr[\Omega] = \sum_{\omega \in \Omega} \Pr[\{\omega\}] = 1$$

     Absolutní suma

     –
     $$\boxed{\forall A \in \mathcal{F}\colon \ \Pr[A] = \sum_{\omega \in A} \Pr[\{\omega\}]}$$

     $(1-p)^3 \cdot p$

     OOOH

     a tento výraz je dobře definovaný (vzpomeňte na absolutní konvergenci, neklesající posloupnost a omezenou posloupnost).

   - *Pravděpodobnost:* Jak jsme viděli v předchozím bodě, tak stačí určit pravděpodobnost, že hodíme právě n-krát, což je

     $$\boxed{\Pr[\{n\}] = (1 - p)^{n-1} p} \qquad \text{(pro libovolné } n \in \mathbb{N}^+\text{)}$$

     protože napřed musí padnout $n - 1$ orlů a pak jedna hlava.

     Zkontrolujme ještě, že se vše sečte na jedničku. Pro jistotu napřed zopakujme součty geometrické řady.

     $$\begin{aligned} S &= \sum_{j=0}^{n} q^j \\ &= 1 + q + q^2 + \ldots + q^n \\ &= 1 + q\left(1 + q + q^2 + \ldots + q^{n-1}\right) \\ &= 1 + q(S - q^n) \end{aligned}$$

     tedy

     $$S = 1 + q(S - q^n)$$

$$S - qS = 1 - q^{n+1}$$
$$S = \frac{1 - q^{n+1}}{1 - q} \qquad \text{(pokud } q \neq 1\text{)}$$

a pro nekonečný případ

$$\begin{aligned}
\sum_{j=0}^{\infty} q^j &= \lim_{n\to\infty} \sum_{j=0}^{n} q^j \\
&= \lim_{n\to\infty} \frac{1 - q^{n+1}}{1 - q} \\
&= \frac{1}{1 - q} && \text{(pokud } |q| < 1\text{)}
\end{aligned}$$

Teď už můžeme aplikovat předchozí pro naší pravděpodobnost:

$$\begin{aligned}
\sum_{n\in\Omega} \Pr[\{n\}] &= \sum_{n\in\Omega} (1-p)^{n-1} p \\
&= p\left(1 + (1-p) + (1-p)^2 + \dots\right) \\
&= 1
\end{aligned}$$

(b) Jaká je pravděpodobnost, že hodíme právě třikrát (n-krát)?

Řešení: To už jsme určili v předchozím bodě, ale tato vlastnost je tak důležitá, že to radši zopakujeme:

n=3

$$\Pr[\{n\}] = (1-p)^{n-1} p \qquad \text{(pro libovolné } n \in \mathbb{N}^+\text{)}$$

2

Připomeňme, že tomuto se také někdy říká geometrické rozdělení. Dejte pozor na to, že $0 < p \leq 1$ (proč?). Hodí se, když při hodu kostkou házíme znovu a zajímá nás celkový počet hodu.

(c) Jaká je pravděpodobnost, že hodíme nejvýš třikrát (n-krát)?

Řešení: Využijeme součet geometrické posloupnosti:

$$\begin{aligned}
\Pr[\{1, 2, \dots, n\}] &= \sum_{j=1}^{n} p(1-p)^{j-1} \\
&= p \sum_{j=1}^{n} (1-p)^{j-1} \\
&= p \frac{1 - (1-p)^n}{1 - (1-p)} \\
&= 1 - (1-p)^n
\end{aligned}$$

3

nepárné

(d) Jaká je pravděpodobnost, že hodíme lišekrát?

Řešení: Opět přímý výpočet:

$$\begin{aligned}
\Pr[\{1, 3, 5, 7, \dots\}] &= \sum_{j=0}^{\infty} p(1-p)^{2j} \\
&= p \sum_{j=0}^{\infty} ((1-p)^2)^j
\end{aligned}$$

$$= p\sum_{j=0}^{\infty}((1-p)^2)^j$$

$$= p\frac{1}{1-(1-p)^2}$$

(e) **Simulujte předchozí.**

Řešení:

```
from random import random


def geometric(pr: float = 0.5) -> int:
    """pr is success probability, return the number of tosses until
    the first success."""
    assert pr > 0
    sample = 1
    fail_pr = 1 - pr   ... padne 0
    while random() < fail_pr:            random() ∈ [0,1)
        sample += 1
    return sample
```

právě 3x / N

```
N = 1000000   # Pokusů
pr = 0.3

exactly_three_sim = sum(int(geometric(pr) == 3) for _ in range(N)) / N
exactly_three = pr * (1 - pr)**2   vzoreček
print(f'a) Pr[tři] = {exactly_three_sim} (= {exactly_three})')

at_most_three_sim = sum(int(geometric(pr) <= 3) for _ in range(N)) / N
at_most_three = 1 - (1 - pr)**3
print(f'b) Pr[nejvýš tři] = {at_most_three_sim} (= {at_most_three})')

odd_number_sim = sum(int(geometric(pr) % 2 == 1) for _ in range(N)) / N
odd_number = pr / (1 - (1 - pr)**2)
print(f'c) Pr[lišekrát] = {odd_number_sim} (= {odd_number})')

# Možný výstup:
# a) Pr[tři] = 0.147168 (= 0.14699999999999996)
# b) Pr[nejvýš tři] = 0.65664 (= 0.657)
# c) Pr[lišekrát] = 0.58815 (= 0.5882352941176471)
```

1kč 2kč

2. **Hodíme cinknutou korunou (panna s pravděpodobností $p_1 \in [0,1]$) a cinknutou dvoukorunou (panna s pravděpodobností $p_2 \in [0,1]$). Oba hody jsou na sobě nezávislé.**

   (a) **Určete pravděpodobnostní prostor.**

   Řešení:

   - *Množina elementárních jevů:*
   $$\Omega = \{P_1P_2, P_1O_2, O_1P_2, O_1O_2\}$$
   kde P je panna O orel a index ukazuje na které minci to padlo.
   - *Prostor jevů:*
   $$\mathcal{F} = \mathcal{P}(\Omega)$$
   - *Pravděpodobnost:* určíme znovu jen na jednoprvkových jevech (na obecném jevu suma):
   $$\begin{aligned}\Pr[\{P_1P_2\}] &= p_1p_2\\ \Pr[\{P_1O_2\}] &= p_1(1-p_2)\\ \Pr[\{O_1P_2\}] &= (1-p_1)p_2\\ \Pr[\{O_1O_2\}] &= (1-p_1)(1-p_2)\end{aligned}$$

   (b) **Připomeňte si definici podmíněné pravděpodobnosti (Definice 2.2).**

   (c) **Spočítejte pravděpodobnost** $\Pr[$**na obou padne panna** $\mid$ **na koruně padne panna**$]$**.**

   $P(A)$

   $P(A \cap B)$

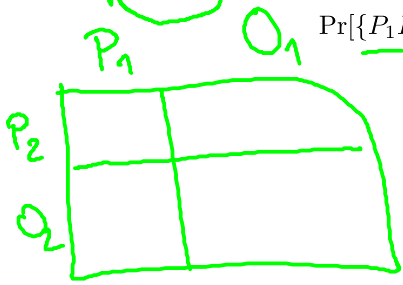
   na přednášce

   $P(A|B)$

   Řešení: Jev A „na obou padne panna“ je formálně $A = \{P_1P_2\}$, jev B „na koruně padne panna“ je formálně $B = \{P_1P_2, P_1O_2\}$ (má pravděpodobnost ostře větší než jedna). Pak podmíněná pravděpodobnost je:
   $$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

   $\Pr[A|B] = \Pr[A]$ nezávislé

   tedy

   $$\begin{aligned}\Pr[\{P_1P_2\} \mid \{P_1P_2, P_1O_2\}] &= \frac{\Pr[\{P_1P_2\} \cap \{P_1P_2, P_1O_2\}]}{\Pr[\{P_1P_2, P_1O_2\}]}\\ &= \frac{\Pr[\{P_1P_2\}]}{\Pr[\{P_1P_2, P_1O_2\}]}\\ &= \frac{p_1p_2}{p_1p_2 + p_1(1-p_2)}\\ &= p_2 \quad \text{(což dává smysl, protože ty hody jsou nezávislé)}\end{aligned}$$

   $\Pr[A|B] \cdot \cancel{\Pr[B]} = \Pr[A \cap B]$

   $= \Pr[A] \cdot \cancel{\Pr[B]}$

   def. nez.

   (d) **Spočítejte pravděpodobnost** $\Pr[$**na obou padne panna** $\mid$ **padne aspoň jedna panna**$]$**.**

   Řešení: Jev A „na obou padne panna“ je formálně $A = \{P_1P_2\}$, jev B „aspoň jedna panna“ je formálně $C = \{P_1P_2, P_1O_2, O_1P_2\}$ (má pravděpodobnost ostře větší než jedna). Pak podmíněná pravděpodobnost je:
   $$\begin{aligned}\Pr[\{P_1P_2\} \mid \{P_1P_2, P_1O_2, O_1P_2\}] &= \frac{\Pr[\{P_1P_2\} \cap \{P_1P_2, P_1O_2, O_1P_2\}]}{\Pr[\{P_1P_2, P_1O_2, O_1P_2\}]}\\ &= \frac{\Pr[\{P_1P_2\}]}{\Pr[\{P_1P_2, P_1O_2, O_1P_2\}]}\end{aligned}$$

$$= \frac{p_1 p_2}{p_1 p_2 + p_1(1 - p_2) + (1 - p_1)p_2}$$

(e) **Simulujte:**

Řešení:

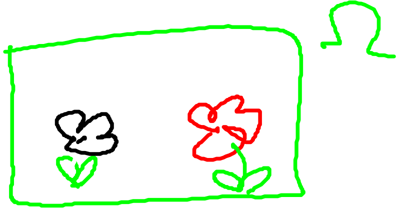
```
from random import random


def toss(weights):→ List[Bool]:
    """True = panna, False = Orel"""
    coins = [False] * len(weights)
    for i in range(len(weights)):
        coins[i] = random() < weights[i]
    return coins


N = 1000000
p1 = 0.1   # panna na první minci
p2 = 0.6   # panna na druhé minci

obe_panna = 0
prvni_je_panna = 0
aspon_jedna_panna = 0

for _ in range(N):
    coins = toss(weights=[p1, p2])
    if all(coins):
        obe_panna += 1
    if coins[0]:
        prvni_je_panna += 1
    if any(coins):
        aspon_jedna_panna += 1

pr_c_sim = obe_panna / prvni_je_panna
pr_c = p2
```

$\Pr[P_1P_2 \mid P_1]$

$\Pr[P_1P_2 \mid P_1 \text{ nebo } P_2]$

```
pr_d_sim = obe_panna / aspon_jedna_panna
pr_d = p1 * p2 / (p1 * p2 + p1 * (1 - p2) + (1 - p1) * p2)

print(f'Pr[obě panna|koruna panna] = {pr_c_sim} (={pr_c})')
print(f'Pr[obě panna|aspoň jedna panna] = {pr_d_sim} (={pr_d})')

# Možný výstup:
# Pr[obě panna|koruna panna] = 0.6017034618418556 (=0.6)
# Pr[obě panna|aspoň jedna panna] = 0.09411977858579801 (=0.09375)
```

Bayes

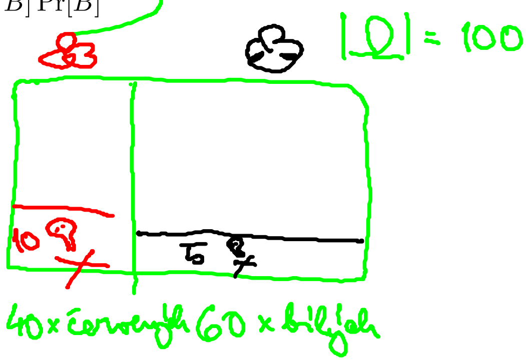


3. **Na louce rostou květiny, které mají buď bílé nebo červené květy. Náhodná květina má bílý květ s pravděpodobností $\Pr[B] = 0.6$, tedy pravděpodobnost že náhodná květina má červený květ je $\Pr[C] = 0.4$. Pravděpodobnost že červená květina je jedovatá je $\Pr[J \mid C] = 0.25$. Pravděpodobnost že bílá květina je jedovatá je $\Pr[J \mid B] = 1/12$. Snědli jsme náhodnou rostlinu a je nám zle, jaká je pravděpodobnost, že ta rostlina měla červený květ?**

Řešení: Řešení 1 – použití Bayesovy věty (Věta 1) jako stroj:

- Máme Ω, což je množina všech květin.
- Máme rozklad Ω (víme, že kvetou buď bíle nebo červeně), tedy jevy $B_1 = B, B_2 = C$.
- Víme $\Pr[J \mid C]$ i $\Pr[J \mid B]$.
- Zajímá nás $\Pr[C \mid J]$.
- Dosadíme do vzorce:

$$\begin{aligned}\Pr[C \mid J] &= \frac{\Pr[J \mid C]\Pr[C]}{\Pr[J \mid C]\Pr[C] + \Pr[J \mid B]\Pr[B]} \\ &= \frac{0.25 \cdot 0.4}{0.25 \cdot 0.4 + (1/12) \cdot 0.6} \\ &= 2/3\end{aligned}$$



Řešení 2 – použijeme představivost a kreslíme:

- Nechť je na louce 100 kytek.
- Takže bílých je 60.
- Červených je 40.
- Jedovatých červených je 10.
- Jedovatých bílých je 5.
- Jedovatých je $\underbrace{15}_{5+10}$ (to je jmenovatel v Bayesově větě).
- Pravděpodobnost že kytka je červená když je jedovatá je $10/15 = 2/3$.

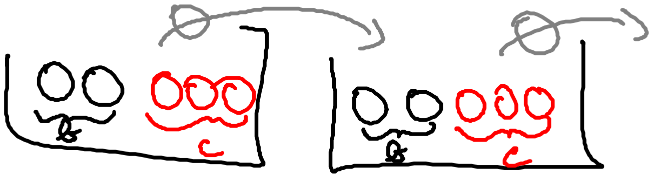
Pár poznámek:

- Čísla byla hezká, takže druhý postup je jednoduchý.
- Naprostá většina lidí nezvládá tento typ úlohy. Takže si nejlépe osvojte oba postupy. První je super pro počítač, druhý pro rychlý odhad.

Můžeme zkusit i naivní simulaci:

```python
from enum import Enum
from random import random


def bernoulli(pr: float = 0.5) -> bool:
    return random() < pr


class Color(Enum):
    RED = 1
    WHITE = 2
```

```
class Plant:
    """Random plant."""
    def __init__(self):
        if bernoulli(0.6):
            self.color = Color.WHITE
            self.is_poisonous = bernoulli(1/12)
        else:
            self.color = Color.RED
            self.is_poisonous = bernoulli(0.25)


N = 1000000  # Pokusů
poisonous = 0  # Kolik jsme viděli jedovatých rostlin.
poisonous_red = 0  # Kolik z těch jedovatých bylo červených

for _ in range(N):
    p = Plant()
    if p.is_poisonous:
        poisonous += 1
        if p.color == Color.RED:
            poisonous_red += 1

assert poisonous > 0, "Pravděpodobnost Pr[A|B] není definovaná pokud Pr[B]=0"
print(f'Viděli jsme {poisonous_red} červených jedovatých rostlin')
print(f'z celkem    {poisonous} jedovatých rostlin')
print(f'tedy Pr[červená|jedovatá] = {poisonous_red/poisonous} (= {2/3})')

# Možný výstup:
# Viděli jsme 100562 červených jedovatých rostlin
# z celkem    150617 jedovatých rostlin
# tedy Pr[červená|jedovatá] = 0.6676669964213867 (= 0.6666666666666666)
```



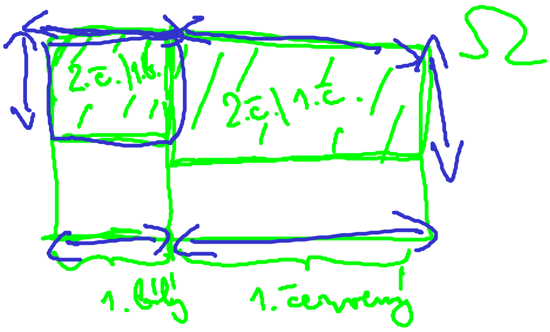
? Pr[2. míček je červený] = ?

4. **V první krabici je b bílých míčků a c červených míčků, ve druhé krabici také. Napřed vytáhneme jeden míček z první krabice (uniformně náhodně) a dáme ho do druhé krabice. Pak vytáhneme jeden míček z druhé krabice (uniformně náhodně). Jaká je pravděpodobnost, že míček vytažený z druhé krabice je červený?**

   (a) **Navrhněte vhodný pravděpodobnostní prostor.**

   Řešení:

   - *Množina elementárních jevů:* $\Omega = \{B_1B_2, B_1C_2, C_1B_2, C_1C_2\}$ kde B_1B_2 značí že jsme z první krabice vytáhli bílý míček (a dali ho do druhé krabice) a pak jsme z druhé krabice vytáhli bílý míček...
   - *Prostor jevů:* $\mathcal{F} = \mathcal{P}(\Omega)$.
   - *Pravděpodobnost:* znovu jen pro jednoprvkové jevy



$$\Pr[\{B_1B_2\}] = \left(\frac{b}{b+c}\right)\left(\frac{b+1}{b+1+c}\right)$$
$$\Pr[\{B_1C_2\}] = \left(\frac{b}{b+c}\right)\left(\frac{c}{b+1+c}\right)$$
$$\Pr[\{C_1B_2\}] = \left(\frac{c}{b+c}\right)\left(\frac{b}{b+1+c}\right)$$
$$\Pr[\{C_1C_2\}] = \left(\frac{c}{b+c}\right)\left(\frac{c+1}{b+1+c}\right)$$

   (b) **Spočítejte tu pravděpodobnost, kterou jsme chtěli.**

   Řešení: Použijeme podmíněnou pravděpodobnost (a projdeme možnosti, co se stane).

$$\begin{aligned}
\Pr[2.\ \text{červený}] &= \Pr[2.\ \text{červený} \mid 1.\ \text{bílý}]\Pr[1.\ \text{bílý}] + \Pr[2.\ \text{červený} \mid 1.\ \text{červený}]\Pr[1.\ \text{červený}] \\
&= \frac{c}{b+1+c}\frac{b}{b+c} + \frac{c+1}{b+1+c}\frac{c}{b+c} \\
&= \frac{bc}{(b+c)(b+1+c)} + \frac{c(c+1)}{(b+c)(b+1+c)} \\
&= \frac{c(b+c+1)}{(b+c)(b+1+c)} \\
&= \frac{c}{b+c}
\end{aligned}$$

Jednodušit!

   A to samé bychom dostali i přímo z rozepsání:

$$\begin{aligned}
\Pr[\{B_1C_2, C_1C_2\}] &= \left(\frac{b}{b+c}\right)\left(\frac{c}{b+1+c}\right) + \left(\frac{c}{b+c}\right)\left(\frac{c+1}{b+1+c}\right) \\
&= \dots \\
&= \frac{c}{b+c}
\end{aligned}$$

   (c) **Simulujte.**

   Řešení:

```
import random

bilych = 15
```

```
cervenych = 37

kbelik_1 = ['B'] * bilych + ['C'] * cervenych    = ['B','B',...,'B','C'.......'C']
kbelik_2 = ['B'] * bilych + ['C'] * cervenych

N = 1000000                                  náhodný míček
cervenych_z_druheho = 0
for _ in range(N):
    druhy_kbelik = kbelik_2 + [random.choice(kbelik_1)]
    assert len(druhy_kbelik) == 1 + len(kbelik_1)        TODO
    if random.choice(druhy_kbelik) == 'C':
        cervenych_z_druheho += 1

vysledek = cervenych / (cervenych + bilych)
print(f'Nasimulovali jsme {cervenych_z_druheho/N} (={vysledek})')

# Možný výstup:
# Nasimulovali jsme 0.711212 (=0.7115384615384616)
```

5. **Tento příklad je vymyšlený, zejména čísla nesedí a reálný svět je malinko složitější (tím se ještě budeme zabývat), ale informace v něm nejsou daleko od pravdy. V zemi nám řádí nemoc C.**

- **Prostor elementárních jevů (sample space) Ω jsou všichni občané.**
- **Označíme $C^+ \subseteq \Omega$ množinu všech lidí, kteří dnes mají aktivní nemoc C, označíme $C^- = \Omega \setminus C^+$ zdravé lidi.**
- **Umíme uniformně náhodně samplovat lidi, tedy $\forall \omega \in \Omega$: $\Pr[\{\omega\}] = 1/|\Omega|$ (tady ω je jeden člověk).**

ideál $T^+ = C^+$

- **Test nám pro libovolného člověka odpoví že je člověk zdravý nebo nemocný. Značme $T^+ \subset \Omega$ množinu lidí pro které test odpoví, že jsou nemocní. Značme $T^- = \Omega \setminus T^+$ množinu lidí pro které test odpoví, že jsou zdraví. Ale není to tak jednoduché, v příbalovém letáku testu se píše:**

  Sensitivity: **(true positive)** $\Pr[T^+ \mid C^+] = 0.9$

  Specificity: **(true negative)** $\Pr[T^- \mid C^-] = 0.8$

  z tohoto můžeme odvodit chyby:

  – **False positive = false alarm = type I error**
  $$\Pr[T^+ \mid C^-] = 1 - \Pr[T^- \mid C^-] = 0.2$$
  – **False negative = miss = type II error**
  $$\Pr[T^- \mid C^+] = 1 - \Pr[T^+ \mid C^+] = 0.1$$

- **Provedli jsme jeden test u každého z uniformně náhodně vybraných 50000 lidí a pozitivních testů vyšlo 1000. Tedy předpokládáme, že** $\Pr[T^+] = \frac{1000}{50000} = \frac{1}{50} = 0.02$ **(jak moc je tento předpoklad oprávněný budeme zkoumat nadále).**
- **Zajímá nás** $\Pr[C^+]$? **(vynásobeno 100 nám dá počet nemocných v procentech).**

(a) **V čem se toto liší od reality?**

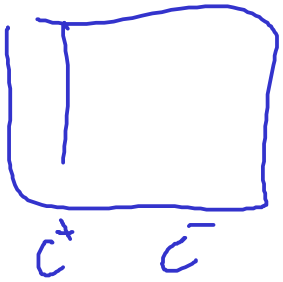
Řešení:

- Zejména v tom náhodném testování. Uvědomte si, že trasování nevybírá lidi náhodně. Ve skutečnosti je velmi těžké vybrat náhodného člověka (více o tom později).
- Sensitivita a specificita jsou opravdu důležité parametry testu (nezávisí na tom jaké je procento nemocných). Jejich výhoda je, že pokud známe četnost nemoci v populaci, pak můžeme pomocí Bayesovy věty spočítat pravděpodobnost, že náhodně vybraný člověk je nemocný, pokud test vyšel pozitivní. Pak se stejně dělá ještě další test, abychom si byli jistí (viz domácí úkol).
- Sensitivita a specificita se určují experimentálně, tedy je neznáme přesně (můžete zkusit v simulaci co to udělá).
- Milion komplikací při popisu skutečného světa, například nevíme na čem závisí pravděpodobnost chyby prvního nebo druhého druhu (třeba je pro danou krevní skupinu false positive pravděpodobnější...).

(b) **Spočítejte** $\Pr[C^+]$**.**

Řešení: Využijeme větu o úplné pravděpodobnosti: pokud $B_1, B_2 \in \mathcal{F}$ je rozklad Ω (tedy $B_1 \cap B_2 = \emptyset$ a navíc $B_1 \cup B_2 = \Omega$, připomeňme že věta platí i pro spočetný rozklad $B_1, B_2, \ldots$), pak pro libovolné $A \in \mathcal{F}$ máme

$$\Pr[A] = \Pr[A \mid B_1] \Pr[B_1] + \Pr[A \mid B_2] \Pr[B_2]$$



0.02 0.9 ? 0.2 = (1 − 0.2)

Aplikujeme předchozí:

$$\begin{aligned}\Pr[T^+] &= \Pr[T^+ \mid C^+]\Pr[C^+] + \Pr[T^+ \mid C^-]\Pr[C^-] \\ &= \Pr[T^+ \mid C^+]\Pr[C^+] + \Pr[T^+ \mid C^-](1 - \Pr[C^+])\end{aligned}$$

?

přeuspořádáme

$$\begin{aligned}\Pr[T^+] &= \Pr[T^+ \mid C^+]\Pr[C^+] + \Pr[T^+ \mid C^-](1 - \Pr[C^+]) \\ \Pr[T^+] &= \Pr[T^+ \mid C^+]\Pr[C^+] + \Pr[T^+ \mid C^-] - \Pr[T^+ \mid C^-]\Pr[C^+] \\ \Pr[T^+] &= (\Pr[T^+ \mid C^+] - \Pr[T^+ \mid C^-])\Pr[C^+] + \Pr[T^+ \mid C^-] \\ \Pr[C^+] &= \frac{\Pr[T^+] - \Pr[T^+ \mid C^-]}{\Pr[T^+ \mid C^+] - \Pr[T^+ \mid C^-]} \\ \Pr[C^+] &= \frac{0.02 - 0.2}{0.9 - 0.2} \\ \Pr[C^+] &\approx -0.257\end{aligned}$$

(c) Co se stalo špatně?

Řešení: Taková data bychom nečekali ani kdyby všichni byli zdraví (vyšlo nám příliš málo pozitivních). Takže jsme rozhodně netestovali náhodný vzorek populace.

(d) Jak by vyšlo předchozí kdyby $\Pr[T^+ \mid C^+] = 0.99, \Pr[T^- \mid C^-] = 0.98, \Pr[T^+] = 0.2$?

Řešení:

$$\begin{aligned}\Pr[C^+] &= \frac{\Pr[T^+] - \Pr[T^+ \mid C^-]}{\Pr[T^+ \mid C^+] - \Pr[T^+ \mid C^-]} \\ &= \frac{0.2 - 0.02}{0.99 - 0.02} \\ &\approx 0.185\end{aligned}$$

Což dává smysl (je spíš pravděpodobné, že zdravého chybně označíme za nemocného než naopak).

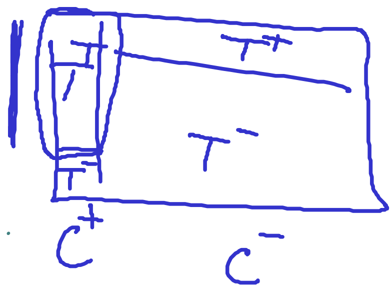
(e) Simulujte předchozí.

Řešení:

```python
from random import random


def bernoulli(pr: float = 0.5) -> bool:
    return random() < pr


class Human:
    """ _illness_probability is our unknown! """
    _illness_probability = 0.185
    """Random human."""
    def __init__(self):
        self.is_ill = bernoulli(Human._illness_probability)
```

? $\in \{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$

```
class IllnessTest:
    sensitivity = 0.99  # = Pr[T+|C+]  } známé
    specificity = 0.98  # = Pr[T-|C-]

    def test(h: Human) -> bool:
        """Return True if the test says h is ill."""
        if h.is_ill:
            return bernoulli(IllnessTest.sensitivity)
        else:
            return not bernoulli(IllnessTest.specificity)
                   not memory!

N = 50000  # Number of samples
false_positives = 0
true_positives = 0
ill_humans = 0

for _ in range(N):
    h = Human()
    if h.is_ill:
        ill_humans += 1
        if IllnessTest.test(h):
            # The test is positive and the human is ill.
            true_positives += 1
    else:
        if IllnessTest.test(h):
            # The test is positive and the human is healthy.
            false_positives += 1

pr_positive_test = (true_positives + false_positives) / N
pr_true_positive = true_positives / ill_humans
pr_false_positive = false_positives / (N - ill_humans)
illness_estimate = ((pr_positive_test - pr_false_positive)
                    / (pr_true_positive - pr_false_positive))

print(f'Pr[positive test]={pr_positive_test}')
print(f'Pr[false positive]={pr_false_positive} (={1-IllnessTest.specificity})')
print(f'Pr[true positive]={pr_true_positive} (={IllnessTest.sensitivity})')
print(f'Pr[nemoc]={illness_estimate} (={Human._illness_probability})')

# Možný výstup:
# Pr[positive test]=0.1976     0.2
# Pr[false positive]=0.019451249387555l2 (=0.020000000000000018)
# Pr[true positive]=0.9897603485583878 (=0.99)
# Pr[nemoc]=0.1836 (=0.185)
```



6. **V šuplíku mám $b \in \mathbb{N}$ párů bílých, $c \in \mathbb{N}$ párů černých ponožek a $s \in \mathbb{N}$ párů sepraných ponožek. Potřebuju si vytáhnout čtyři páry černých ponožek (jedu na prodloužený víkend tancovat). Když vytáhnu čtyři náhodné páry ponožek (mám je napárované v šuplíku), jaká je pravděpodobnost, že všechny budou černé?**

   Řešení: Dle definice podmíněné pravděpodobnosti (Definice 2.2) můžeme napsat (která pravděpodobnost musí být nenulová?):

$$\Pr[A \cap B] = \Pr[A]\Pr[B \mid A]$$
$$\Pr[A \cap B \cap C] = \Pr[A]\Pr[B \mid A]\Pr[C \mid A \cap B]$$

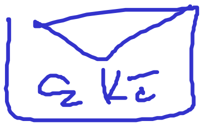
   Tedy můžeme psát C_1, C_2, C_3, C_4 jevy, že první, druhý, třetí, čtvrtý pár vytažených ponožek jsou černé.

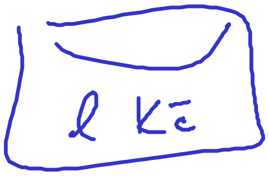
$$\begin{aligned}\Pr[C_1 \cap C_2 \cap C_3 \cap C_4] &= \Pr[C_1]\Pr[C_2 \mid C_1]\Pr[C_3 \mid C_1 \cap C_2]\Pr[C_4 \mid C_1 \cap C_2 \cap C_3] \\ &= \left(\frac{c}{b+c+s}\right)\left(\frac{c-1}{b+c-1+s}\right)\left(\frac{c-2}{b+c-2+s}\right)\left(\frac{c-3}{b+c-3+s}\right)\end{aligned}$$

   Mohli bychom spočítat kolik je čtveřic černých ze všech čtveřic (ale předchozí postup bývá užitečný):

$$\Pr[C_1 \cap C_2 \cap C_3 \cap C_4] = \frac{\binom{c}{4}}{\binom{b+c+s}{4}}$$

```python
from random import sample
from scipy.special import comb

b = 10   # bilych ponozek
c = 15   # cernych ponozek
s = 5    # sepranych ponozek

suplik = ['B'] * b + ['C'] * c + ['S'] * s

N = 1000000
cernych = 0
for _ in range(N):
    vyber = sample(suplik, k=4)
    if vyber == ['C'] * 4:
        cernych += 1

pr_vsechny_cerne = c*(c-1)*(c-2)*(c-3) / ((b+c+s)*(b+c-1+s)*(b+c-2+s)*(b+c-3+s))
print(f'Pr[4 cerne] = {cernych/N} (={pr_vsechny_cerne})')

pr_vsechny_cerne_komb_cislo = comb(c, 4, exact=True) / comb(b+c+s, 4, exact=True)
assert abs(pr_vsechny_cerne - pr_vsechny_cerne_komb_cislo) < 0.000001

# Možný výstup:
# Pr[4 cerne] = 0.049479 (=0.04980842911877394)
```





7. **Na stole jsou dvě obálky, v jedné je k korun, ve druhé ℓ korun ($k, \ell \in \mathbb{N}$). Můžete otevřít jednu obálku a na základě sumy v ní se rozhodnout jestli si necháte tu otevřenou nebo si vezmete tu druhou (nehledě na to, kolik je v té druhé). Umíte vymyslet způsob jak odejít s tou s větším obnosem s pravděpodobností ostře větší než jedna polovina?**

Řešení: Uniformně náhodně zvolíme první obálku. Pokud vidíme m korun, pak házíme spravedlivou mincí, dokud nepadne hlava. Pokud celkový počet hodů je ostře menší než m, pak si obálku necháme, jinak si vezmeme tu druhou. Když $k < \ell$ tak pravděpodobnost, že vyměníme obálku s k korunami je ostře větší než pravděpodobnost, že vyměníme obálku s ℓ korunami.

```python
from random import randint
from random import random


def geometric(pr: float = 0.5) -> int:
    """pr is success probability, return the number of tosses until
    the first success."""
    assert pr > 0
    sample = 1
    fail_pr = 1 - pr
    while random() < fail_pr:
        sample += 1
    return sample


# Our unknown amounts.
envelopes = [5, 10]

N = 1000000       # Number of samples.
total_amount = 0  # Total sum that we got during all samples.
got_larger = 0    # Number of times we walked away with the larger sum.

for _ in range(N):
    # Pick the first envelope at random.
    chosen = randint(0, 1)
    if geometric() < envelopes[chosen]:
        # Keep this one.
        pass
    else:
        # Choose the other.
        chosen = 1 - chosen
    if envelopes[chosen] >= envelopes[1 - chosen]:
        got_larger += 1
    total_amount += envelopes[chosen]

print(f'Pr[selected larger] = {got_larger / N}')
print(f'E[win] = {total_amount / N}')

# Possible outcome:
# Pr[selected larger] = 0.530087
# E[win] = 7.650435
```