

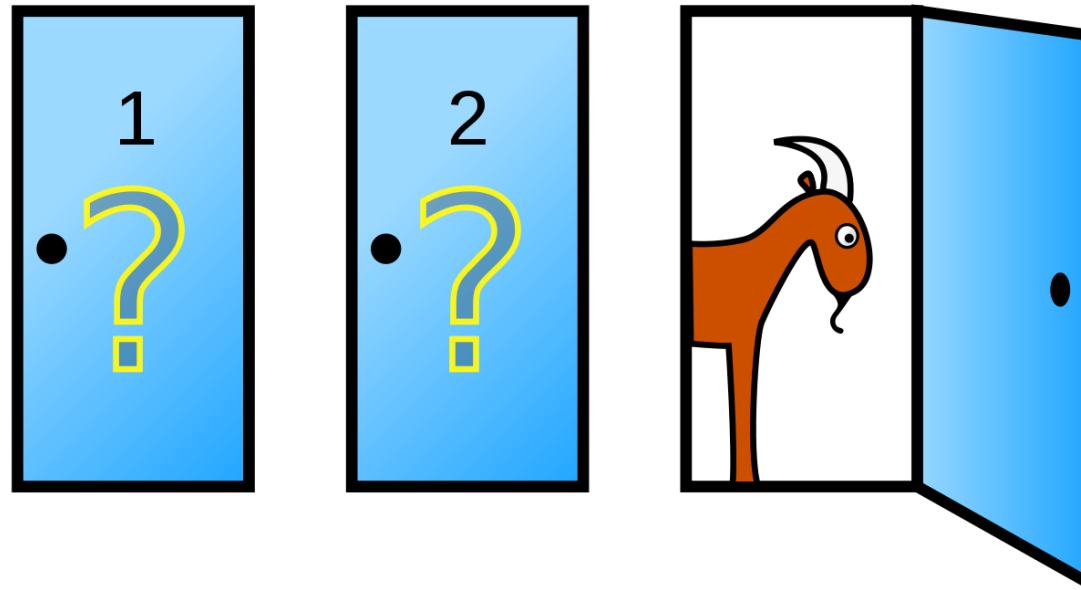
Programovanie v jazyku Python

Jednoduché simulácie
prednáška 9

Katedra kybernetiky a umelej inteligencie
Technická univerzita v Košiciach
Ing. Ján Magyar, PhD.

Monty Hall Problem

- V televíznej súťaži sú pred nami tri dvere, za jednými je hlavná cena (auto), za zvyšnými stojí koza. Vyberieme si niektoré dvere a následne nám otvorí jedny dvere, za ktorými je koza. Následne nám moderátor dá možnosť zmeniť náš výber.
- Otázka: Mali by sme zmeniť rozhodnutie? Zvýšime tak našu šancu na výhru?



- Marilyn vos Savant
ak zmeníme vybrané dvere,
šanca na výhru je 67%, inak 33%



- “You blew it, and you blew it big! Since you seem to have difficulty grasping the basic principle at work here, I'll explain. After the host reveals a goat, you now have a one-in-two chance of being correct. Whether you change your selection or not, the odds are the same. There is enough mathematical illiteracy in this country, and we don't need the world's highest IQ propagating more. Shame!”

- Scott Smith, PhD.
University of Florida

Narodeninový paradox

- Máme v miestnosti n študentov, a zaujíma nás pravdepodobnosť toho, že (aspoň) dvaja majú narodeniny v rovnaký deň.
- Pri akej hodnote n bude pravdepodobnosť spoločných narodenín väčšia ako 50%?
- Pri akej hodnote n bude pravdepodobnosť spoločných narodenín väčšia ako 99%?
- Pri akej hodnote n bude pravdepodobnosť spoločných narodenín väčšia ako 99,9%?

Paradox blesku

V istom meste udrie blesk v priemere raz za 30 dní, každý deň s rovnakou pravdepodobnosťou. Dnes blesk udrel. V ktorý deň udrie nabudúce s najväčšou pravdepodobnosťou?

Modelovanie a simulácia

- cieľom je nájsť riešenie na problémy, ktoré nevieme analyticky popísať
- často nedokážeme otestovať všetky možné výstupy, simulujeme iba reprezentatívnu vzorku
- výsledok je deskriptívny, nie preskriptívny
- simulácia nám nedá riešenie problému, iba ho popíše

Postup pri simuláciach

1. formalizácia problému - určiť otázku, na ktorú chceme odpovedať
2. formalizácia hypotézy - čo očakávame
3. definícia abstrakcie a vytvorenie výpočtových modelov
4. (spracovanie údajov)
5. vykonanie simulácií
6. vyhodnotenie výsledkov
7. vyhodnotenie kvality odpovede

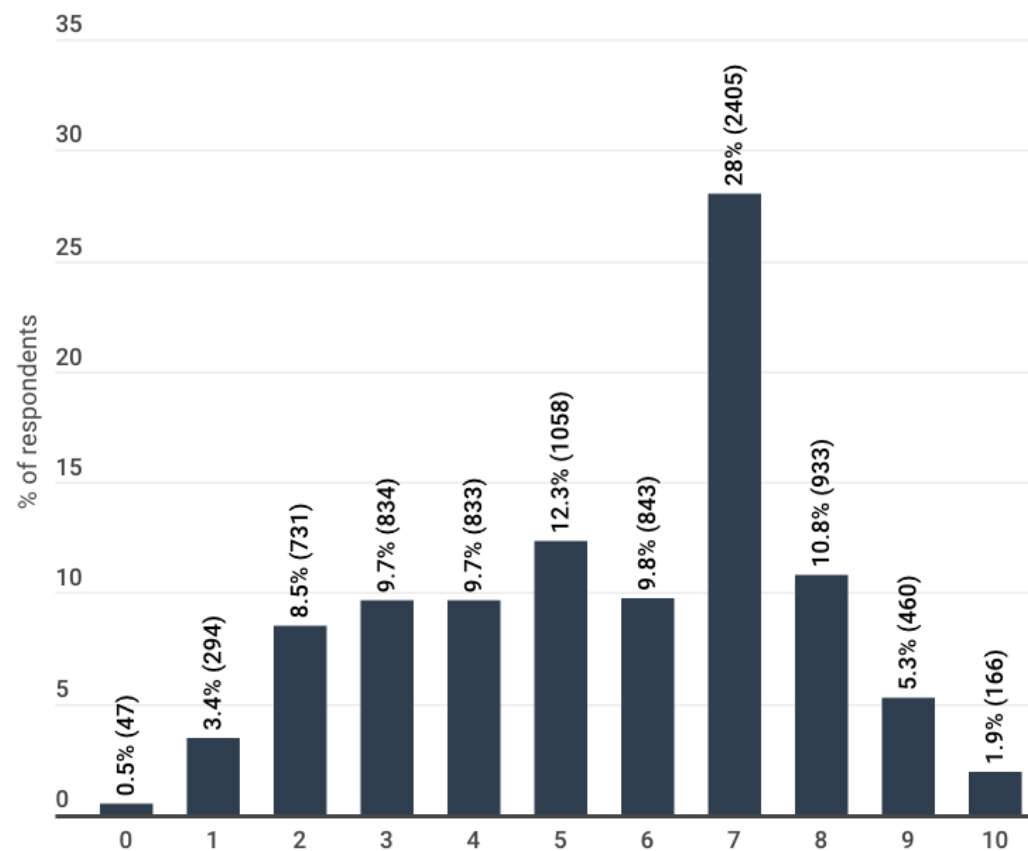
Výpočtový model

- programová reprezentácia namodelovaného javu, resp. sveta
- samotný kód nie je až taký dôležitý
- cieľom je namodelovať reálny svet, mali by sme začať s jednoduchým modelom, iteratívne ho rozširovať podľa potreby
- vo výpočtových modeloch zvyčajne používame náhodnosť - stochastické modely
- základný problém s náhodnosťou: počítače sú deterministické stroje, ako môžeme pomocou nich vygenerovať náhodné čísla?

Dokážu ľudia generovať náhodné čísla?

Pick a random number from 1-10

($n=8604$, mean=5.687, median=6)



Náhodnosť

- má zmysel hovoriť iba o postupnosti náhodných čísel
- počítače generujú pseudonáhodné čísla - čísla sa zdajú byť náhodne generované, ale v skutočnosti ich dostaneme pomocou deterministického algoritmu
- vstupom je zvyčajne čas (v milisekundách), alebo teplota procesora, atď.
- dajú sa generovať aj náhodné čísla - vyžadujú vstup od používateľa (napr. šifrovací kľúč)

Náhodnosť v Pythone

- metódy implementované v module `random`
- defaultne založené na systémovom čase (v ms)
- môže použiť OS-špecifické zdroje náhodnosti
- `random.seed(a=None, version=2)`
 - inicializácia generátora náhodných hodnôt
 - `a` - parameter generovania
 - `version` - rozsah seed hodnôt
 - podpora reprodukovania výsledkov

Generovanie celých čísel

- `random.randint(a, b)`
 - náhodné celé číslo N , kde $a \leq N \leq b$
- `random.randrange([start,] stop[, step])`
 - náhodný prvok z `range(start, stop, step)`
 - reálne sa nevygeneruje objekt typu `range`
 - viac uniformné rozdelenie, ako `int(random.random() * n)`

Náhodnosť so sekvenciami

- `random.choice(seq)`
 - vyberie náhodný prvok zo sekvencie
- `random.choices(population, weights=None, cum_weights=None, k=1)`
 - vyberie k prvkov z populácie a vráti ich v jednom zozname
 - `weights/cum_weights` - definuje pravdepodobnosť výberu jednotlivých prvkov
- `random.shuffle(x[, random])`
 - pomieša prvky sekvencie v náhodnom poradí
 - ak sekvencia je nemenná, je možné zavolať metódu `sample`
- `random.sample(population, k)`
 - vyberie k jedinečných prvkov z populácie
 - vracia nový zoznam s hodnotami

Generovanie desatinných čísel

- `random.random()`
 - náhodné desatinné číslo z rozsahu $[0.0, 1.0)$
- `random.uniform(a, b)`
 - náhodné desatinné číslo z rozsahu $[a, b]$ alebo $[b, a]$
- `random.gauss(mu, sigma)`
 - náhodné číslo z normálneho rozdelenia
 - `mu` - najpravdepodobnejšia hodnota
 - `sigma` - smerodajná odchýlka
- `random.expovariate(lambd)`
 - náhodné číslo z exponenciálneho rozdelenia
 - čísla sú z rozsahu $[0, \infty)$ ak `lambd > 0`, $(-\infty, 0]$ ak `lambd < 0`

Reprezentácia pravdepodobnosti v Pythone

- Ako implementujeme, ak jav sa má stať s 80% pravdepodobnosťou?

```
if random.random() < 0.8:  
    # simulate phenomenon
```

Zhrnutie

- modelovanie a simulácia
- postup pri simuláciách
- výpočtový model
- základné funkcie v module random
- reprezentácia pravdepodobnosti