



Programovanie v jazyku Python

Výpočtové modely a simulácie
prednáška 7

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

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 simuláciách

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
- 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?

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`

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?

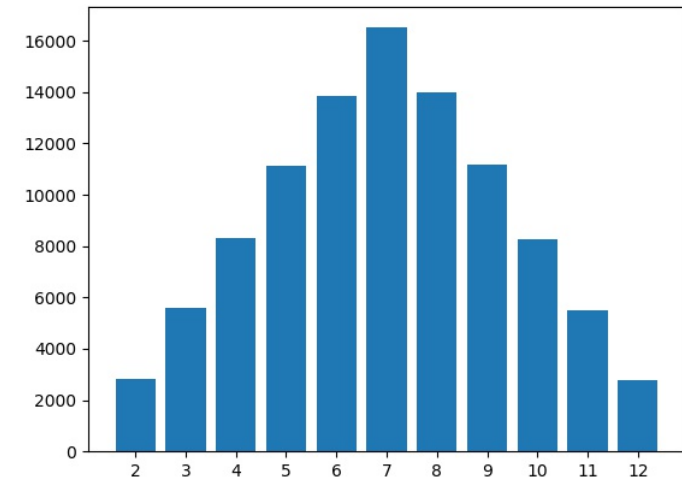
```
if random.random() < 0.8:  
    # simulate phenomenon
```

Uniformné rozdelenie

- každá hodnota (alebo skupina hodnôt) je reprezentovaná v rovnakom pomere v populácii
- každá hodnota bude výsledkom pokusu s rovnakou pravdepodobnosťou
- rozdelenie vieme popísať pomocou intervalu (najmenšia a najväčšia hodnota)
- nie je časté v skutočnosti, iba v modeloch

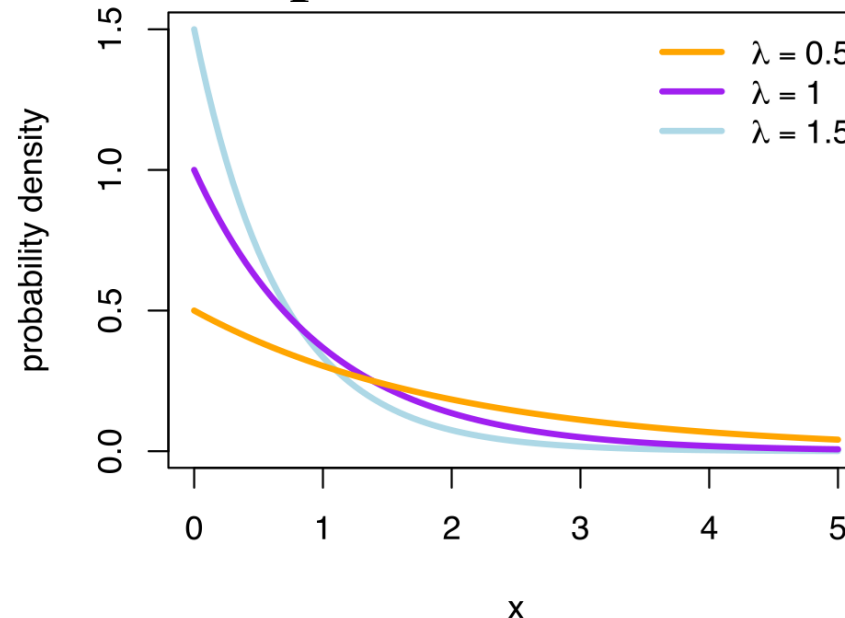
Normálne rozdelenie

- Gaussovo rozdelenie
- stabilné rozdelenie - dostaneme približne rovnaký výsledok aj pri viacerých pokusoch
- najpravdepodobnejšie je priemerná hodnota (súčasne aj medián)
- rozdelenie vieme popísať pomocou priemeru a smerodajnej odchýlky
- asi najčastejšie rozdelenie v prírode



Exponenciálne rozdelenie

- nesymetrické rozdelenie
- najpravdepodobnejšia je najmenšia hodnota
- vieme ho popísať konštantou λ
- vyjadruje rozdelenie uplynutého času medzi dvoma inštanciami toho istého javu, ktorý nastane s konštantou priemernou frekvenciou (doprava, návštevnosť web stránky)



Random walk

Sú tri hodiny v noci, párty skončila, a študenti pomaly odchádzajú na svoje izby. Vzhľadom ale na to, že párty bola veľmi úspešná, každý z nich má značne limitované kognitívne a motorické schopnosti. Jeden z vašich kamarátov sa zabával až tak dobre, že jeho pohyb nevieme nazvať inak, ako náhodný. Keďže vám záleží na ňom, chcete ho nájsť pred tým, ako odídete na izbu. Naposledy ste ho videli pred piatimi minútami.

- Kde by ste ho mali začať hľadať?
- Bude viac vzdialený od bodu, kde ste ho videli naposledy, ak čakáte dlhšie?

Analytický popis

- V akej vzdialenosti bude po jednom kroku?
- V akej vzdialenosti bude po dvoch krokoch?
- V akej vzdialenosti bude po troch krokoch?
- ...
- V akej vzdialenosti bude po n krokoch?

Riešenie výpočtovým modelom

- mapu reprezentujeme pomocou dvojrozmerného priestoru
- nepredpokladáme žiadne prekážky
- každý krok má rovnakú dĺžku
- váš kamarát každú sekundu urobí jeden krok

Štruktúra simulácií

- vnútorný tok (inner loop), ktorý nám vykoná jeden pokus
- vonkajší tok (outer loop), ktorý vykoná niekoľko pokusov (aký je vhodný počet?)
- vypočítame štatistické charakteristiky - chceme získať čo najviac pohľadov, aby sme dostali celkový obraz
- výsledky vizualizujeme vhodným spôsobom

Random walk

- často používaný model, ktorý vie popísať rôzne javy
- Brownov pohyb - pohyb častíc, molekúl, atď.
- modelovanie biologických procesov, evolúcie, pohybu zvierat
- modelovanie signálov v mozgu
- stávkovanie
- aproximácia veľkosti internetu
- modelovanie burzy (nikto nevie, či je vhodná metóda)

Typy simulací

- stochastická vs. deterministická
- statická vs. dynamická
- diskrétna vs. spojitá

Simulácia Monte Carlo

- inferenčná štatistická metóda
- náhodne vybraná vzorka zvyčajne má rovnaké vlastnosti, ako celková populácia
- vzorka musí byť dostatočne veľká
- nie každá vzorka istej veľkosti zodpovedá populácii

Veľkosť vzorky

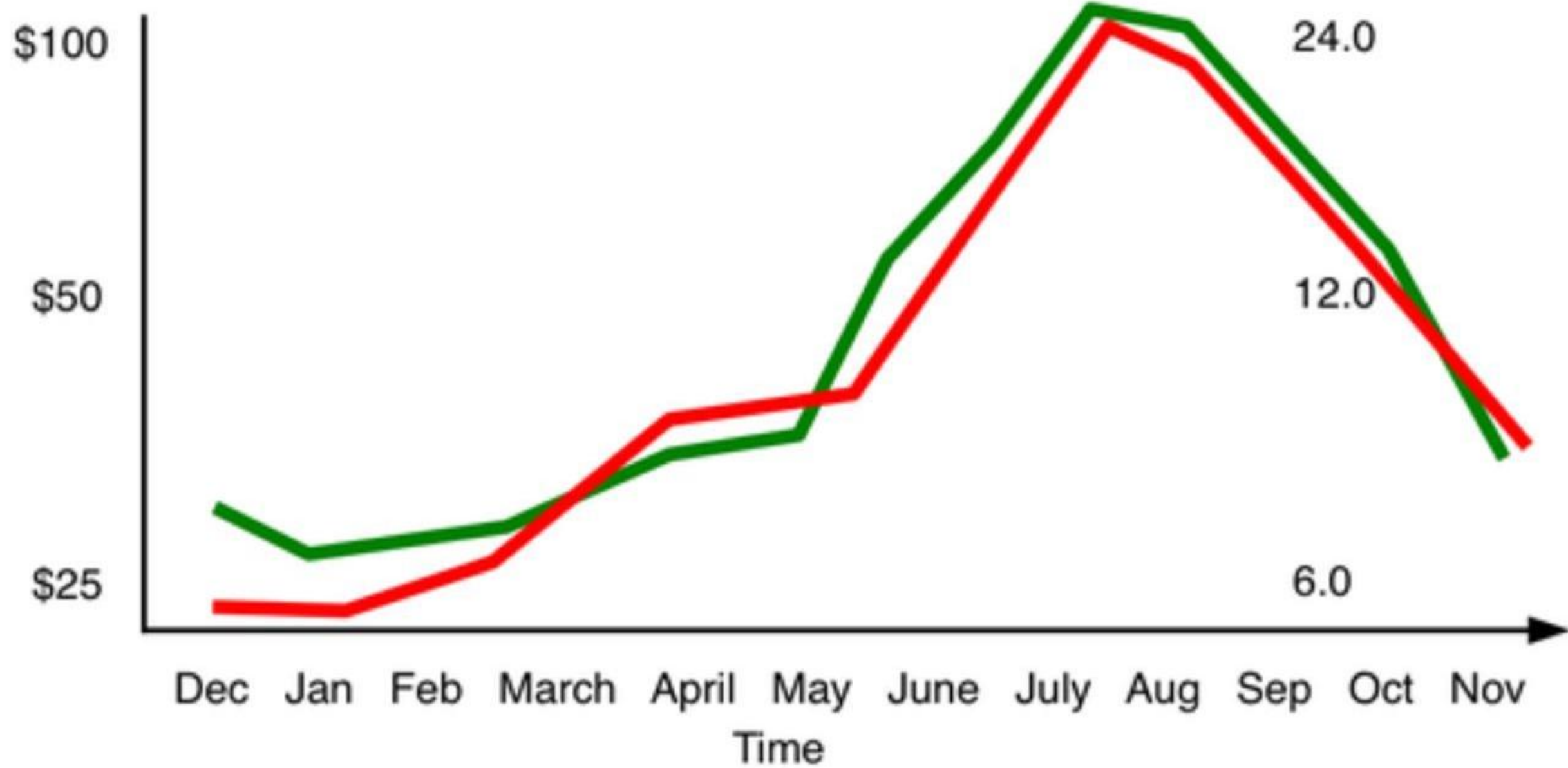
- pri simuláciách je kľúčový problém určenie potrebného počtu iterácií a potrebnej veľkosti vzoriek
- s reprezentatívnymi vzorkami nikdy nedosiahneme 100% presnosť
- šancu na nevyrovnanú vzorku vieme eliminovať len ak uvažujeme celú populáciu
- distribúcia hodnôt vo vzorke by mala byť približne rovnaká, ako distribúcia hodnôt v celej populácii

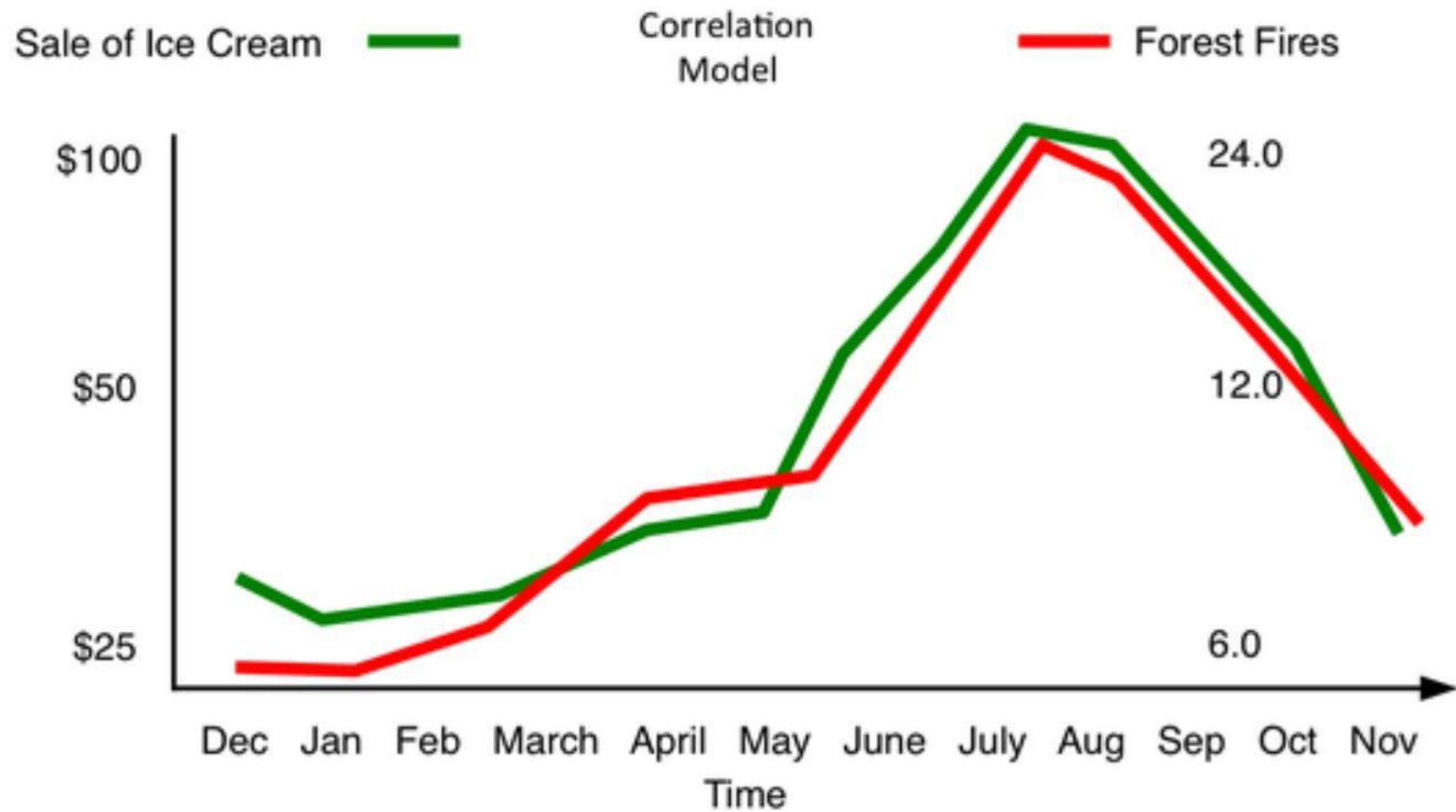
Vyhodnotenie výsledkov

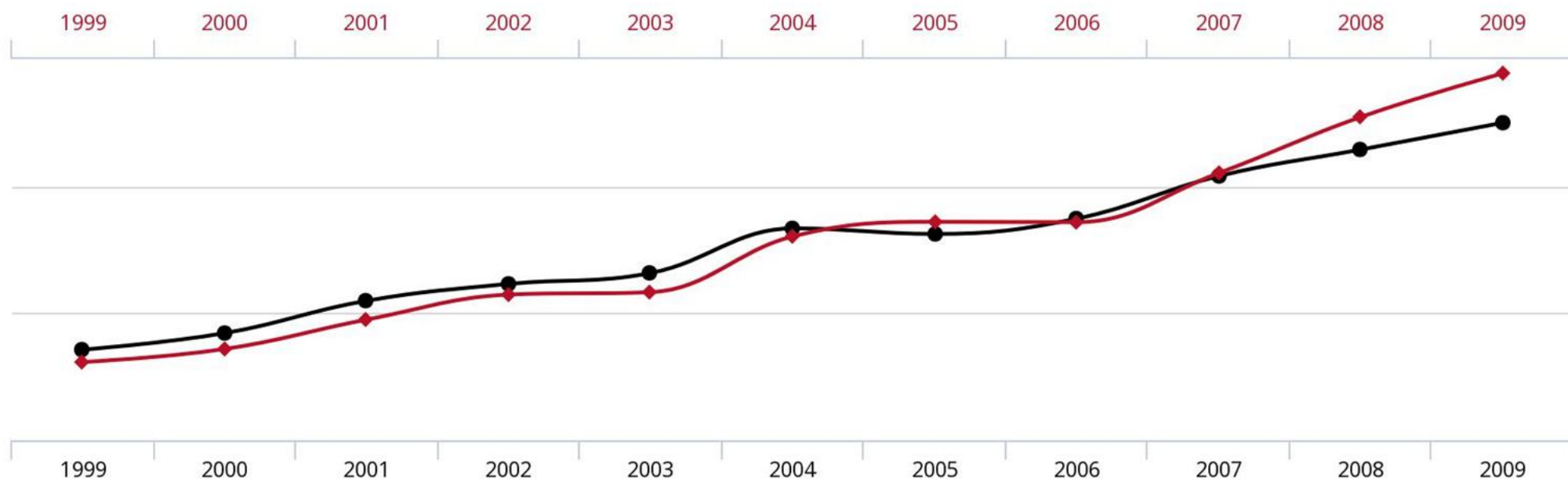
- Odkiaľ vieme, že výsledok simulácie zodpovedá realite?
- je potrebné otestovať implikácie výsledku na reálnom príklade (napr. aproximácia π)
- výstupom simulácie sú
 - údaje
 - modely
 - implikácie alebo následky

Vyhodnotenie výsledkov

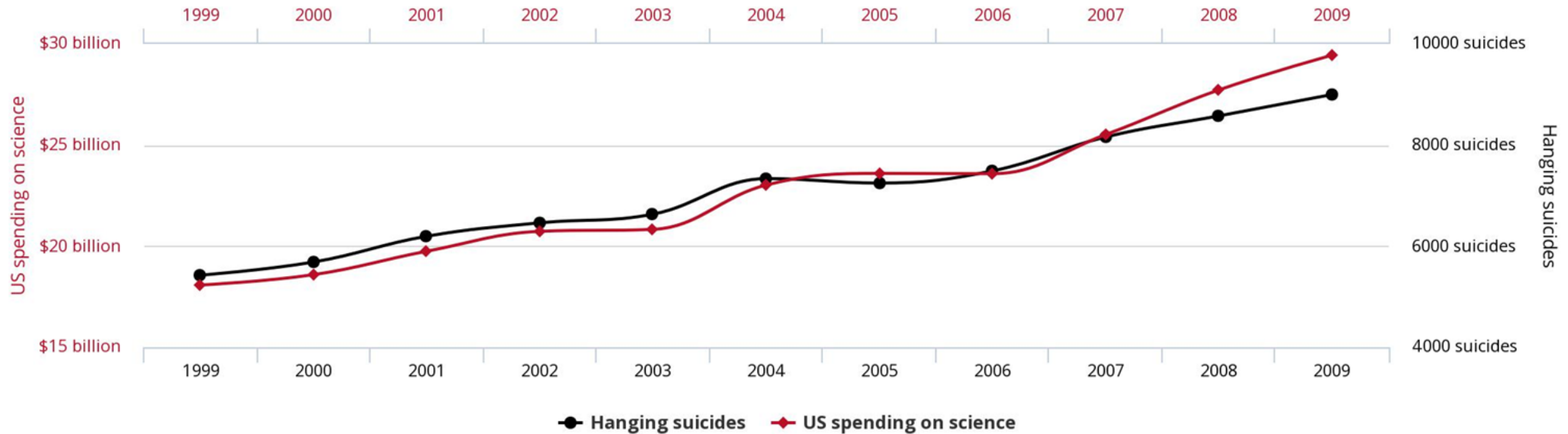
1. je potrebné analyzovať údaje, nielen ich vlastnosti!
napr.: priemer získaných bodov na teste
2. máme reprezentatívnu vzorku? AKA data enhancement
cieľom je otestovať, a nie potvrdiť hypotézu
3. paradox texaského strelca
neignorujeme časť údajov
4. uponáhľané závery
napr.: vyššia nehodovosť v okolí domu
5. korelácia \neq kauzalita







US spending on science, space, and technology correlates with Suicides by hanging, strangulation and suffocation



Craps / Kocky

Craps je populárna kocková hra v kasínach, kde hráč hodí dvomi kockami naraz a vyhrá podľa nasledovných pravidiel:

- ak súčet čísel je 7 alebo 11, vyhrá
- ak súčet čísel je 2, 3 alebo 12, prehrá
- v ostatných prípadoch súčet čísel sa stane cieľovým počtom bodov, a hráč hodí kockami dovtedy, kým nehodí
 - 7, v tomto prípade prehrá
 - cieľový počet bodov, v tomto prípade vyhrá

Akú šancu má hráč na výhru? Má kasíno väčšiu šancu?

Craps / Kocky

- Ako zvýšime šancu výhry hráča?
- Ako zvýšime šancu výhry kasína?

Zhrnutie

- všeobecná štruktúra simulácií
- typy simulácií
- random walk simulácia
- Monte Carlo simulácie
- štatistické rozdelenia
- vyhodnotenie výsledkov
- chyby pri vyhodnocovaní výsledkov