Náhodná procházka

Představte si, že se budeme se pohybovat po přímce následujícím způsobem. Začneme v počátku soustavy souřadnic. Před každým krokem si hodíme korunou. Když padne orel uděláme krok vlevo, když padne pana uděláme krok vpravo. Celkem uděláme N kroků.

Vypočítejte

1. Jaká bude střední vzdálenost od počátku po N krocích?
2. Jaká je pravděpodobnost, že se po N krocích vrátím zpátky do počátku?

Ověřte si výpočet tím, že náhodnou procházku nasimulujete buď v Pythonu, nebo v Excelu. Nasimulujte 100 náhodných procházek o N krocích a nakreslete

1. závislost vzdálenosti od počátku na počtu kroků
2. Závislost pravděpodobnosti, že se vrátím do počátku na počtu kroků

Náhodná procházka

1. Jaká bude střední vzdálenost od počátku po N krocích?

- Předpokládejme, že krok má velikost 1
- Označme to, že padla panna jako úspěch, pravděpodobnost, že v jednom hodu padne panna je p = 0.5
- Pokud padla při N opakováních k-krát panna, tj. k-krát nastal úspěch, dostal jsem se do polohy

$k-(N-k)=2k-N$ (to může být kladné nebo záporné)

- vezmu kvadrát vzdálenosti (aby to bylo vždy kladné) $(2k-N)^2$
- Počet úspěchů k při N opakováních se řídí binomickým rozdělením B(N,p)
- Střední hodnota kvadrátu vzdálenosti je $E[(2k-N)^2]=\sum_{k=0}^{N}(2k-N)^2\frac{N!}{(N-k)!\,k!}p^k(1-p)^{N-k}$

$$E[(2k-N)^2]=E[4k^2-4kN+N^2]=4N(N-1)p^2+4Np-4N^2p+N^2$$

$$p=\frac{1}{2} \qquad \longrightarrow \qquad E[(2k-N)^2]=N^2-N+2N-2N^2+N^2=N$$

- Střední hodnota vzdálenosti je tedy $\sqrt{N}$

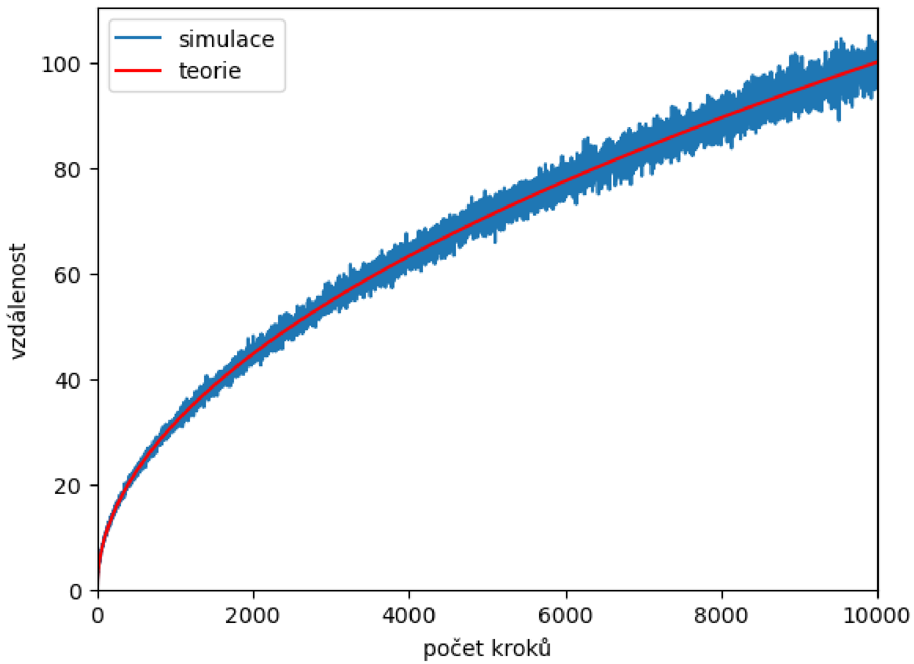
Náhodná procházka

random_walk.py

```
import numpy as np
import matplotlib.pyplot as plt
```

```
Nmax=10000 #maximální počet kroků
p=0.5 #pravděpodobnost, že udělám krok vpravo
N_repeat=1000 #počet opakování simulací
N=np.arange(0,Nmax+1,1) #pole celkových počtů kroků
mean_distance=np.zeros(np.size(N)) #průměrná vzdálenost od středu
mean_distance_abs=np.zeros(np.size(N)) #průměrná vzdálenost od středu
for i in N:
    data=np.random.binomial(N[i],p,N_repeat) #simulace N_repeat hodnot z binomick0ho rozdělení B(N,p)
    mean_distance[i]=np.sqrt(np.mean((2*data-N[i])**2)) #výpočet průměrné hodnoty vzdálenosti od středu jako sqrt(<(2k-N)**2>)
    mean_distance_abs[i]=np.mean(abs((2*data-N[i]))) #výpočet průměrné hodnoty vzdálenosti od středu jako |2k-N|
```

```
plt.plot(N,mean_distance,label='simulace')
plt.plot(N,np.sqrt(N),c='red',label='teorie')
plt.xlim(0,Nmax)
plt.ylim(0,)
plt.xlabel('počet kroků')
plt.ylabel('vzdálenost')
plt.legend()
plt.show()
```

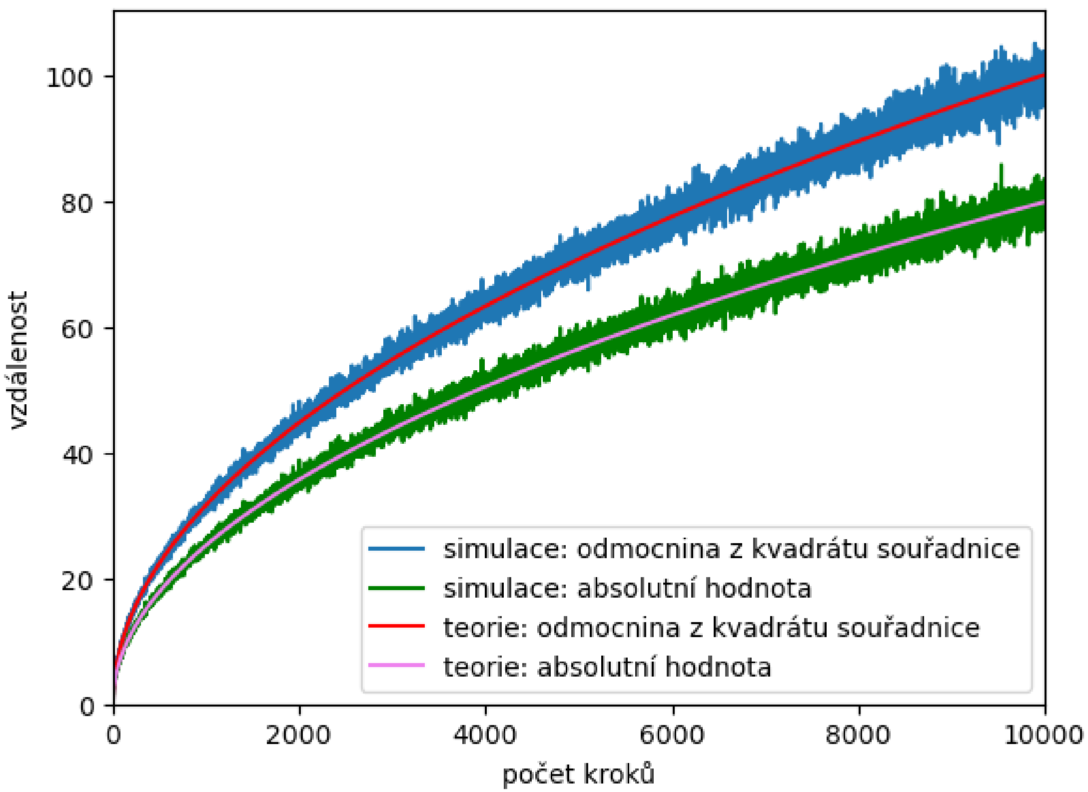


Náhodná procházka random_walk.py

- Pokud počítáme střední hodnotu jako střední hodnotu absolutních hodnot:

$$E[|2k-N|] = \sqrt{\frac{2}{\pi}N}\left(1 \pm \frac{1}{2^2 N} + \frac{1}{2^5 N^2} \pm \frac{5}{2^7 N^3} - \frac{21}{2^{11} N^4} \pm \dots\right)$$

- pro velká N je střední hodnota vzdálenosti: $E[|2k-N|] \approx \sqrt{\frac{2}{\pi}N}$



Náhodná procházka

2. Jaká je pravděpodobnost, že se po *N* krocích vrátím zpátky do počátku?

- Abych se vrátil do počátku, musím udělat stejně kroků vpravo jako vlevo, tj. $k = N / 2$
- Pokud je *N* liché, $P = 0$ (nikdy se nevrátím do počátku)
- Pokud je *N* sudé, spočítáme pravděpodobnost z binomického rozdělení

$$P\left(k = \frac{N}{2} \middle| N, p = \frac{1}{2}\right) = \frac{N!}{\left(\frac{N}{2}!\right)^2} \frac{1}{2^N}$$

Náhodná procházka

P_return.py

```python
import numpy as np
import matplotlib.pyplot as plt
import math
```

```python
Nmax=100#maximální počet kroků
p=0.5 #pravděpodobnost, že udělám krok vpravo
N_repeat=1000 #počet opakování simulací
N=np.arange(0,Nmax+1,1) #pole celkových počtů kroků
Psim=np.zeros(np.size(N)) #pravděpodobnost, že se vrátím do počátku - odhad ze simulace
for i in N:
    data=np.random.binomial(N[i],p,N_repeat) #simulace počtu kroků vpravo pomocí binomického rozdělení
    for k in data:   #počítám kolikrát se vrátím do počátku
        if (2*k-N[i])==0:
            Psim[i]+=1
Psim=Psim/N_repeat
```

```python
Pteor=np.zeros(np.size(N))
for i in N:
    Pteor[i]=math.comb(int(N[i]),int(N[i]/2))*(1/2)**N[i] #pravděpodobnost,
```

```python
plt.step(N,Psim,label='simulace')
plt.plot(N,Pteor,c='red',label='teorie')
plt.xlim(0,Nmax)
plt.ylim(0,1)
plt.xlabel('počet kroků')
plt.ylabel('pravděpodobnost')
plt.legend()
plt.show()
```