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, ROOTu nebo 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 $E[(2k-N)^2] = \sum_{k=0}^{N} (2k-N)^2 \frac{N!}{(N-k)!\,k!} p^k (1-p)^{N-k}$

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

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

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

Náhodná procházka

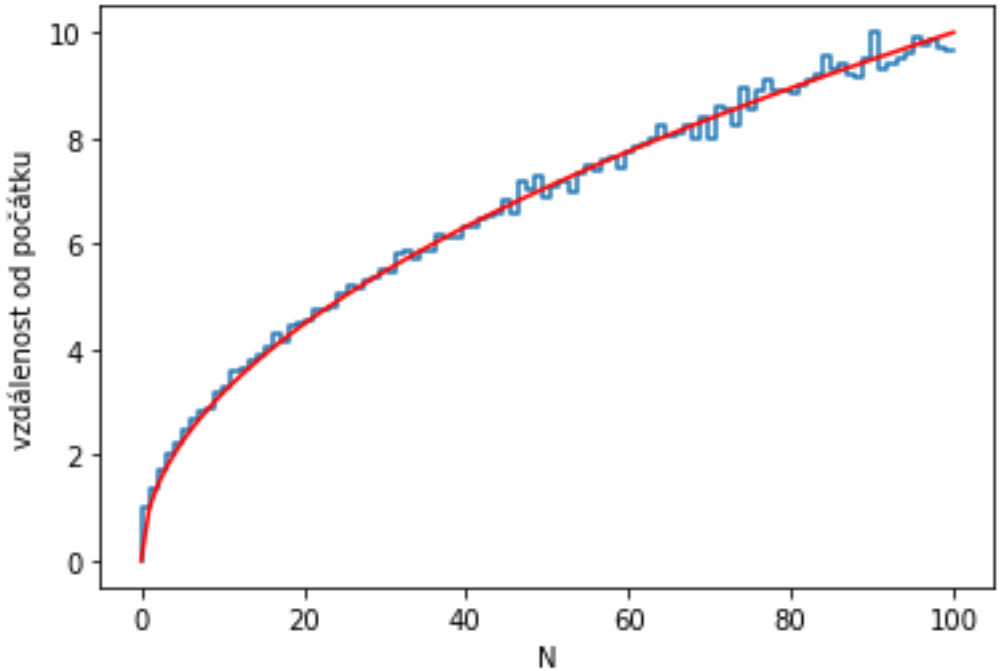
random-walk.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import factorial
Nsim=1000 #pocet nahodnych prochazek, ktere budeme simulo
N=100 #počet kroku budu postupne zvysovat od 0 do N
p=0.5 #pravdepodobnost ze pujdu vpravo
k_data=np.array(Nsim,dtype=int) #pocet kroku vpravo, bino
x_N=np.linspace(0,N,N) #x osa pro vykresleni sqrt(N)
f=np.sqrt(x_N) #odmocninova zavislost
x_N_odd=np.arange(0,N+2,2,dtype=int) #pole sudych cisel N
P_teor=factorial(x_N_odd)/(factorial(x_N_odd/2))**2*(1/2)
distance=np.array(Nsim) #vzdalenosti od pocatku
distance_mean=np.empty(N) #prumerna vzdalenost
P_zero=np.empty(N) #P ze se vratim do nuly

for i in range(0,N): #sumulujeme pocet kroku od 0 do N
    distance_zero_count=0 #pocet pripadu kdy jsem se vrat
    k_data=np.random.binomial(i,p,Nsim) #vygenerujeme Nsi
    distance=(k_data - (i-k_data))**2 #vzdalenost od poca
    for k in range(0,Nsim):
        if distance[k]==0: distance_zero_count=distance_z
    distance_mean[i]=np.sqrt(np.mean(distance)) #prumerna
    P_zero[i]=distance_zero_count/Nsim #P ze se vratim do pocatku

fig,ax=plt.subplots()   #graf zavislosti vzdalenosti od pocatku na N
ax.set_xlabel("N")
ax.set_ylabel("vzdálenost od počátku")
ax.step(x_N,distance_mean)
ax.plot(x_N,f,c="red")

fig,ax=plt.subplots() #graf zavislosti P ze se vratim do pocatku na N
ax.set_xlabel("N")
ax.set_ylabel("Pravděpodobnost návratu do počátku")
ax.step(x_N,P_zero)
ax.plot(x_N_odd,P_teor,c="red")
```



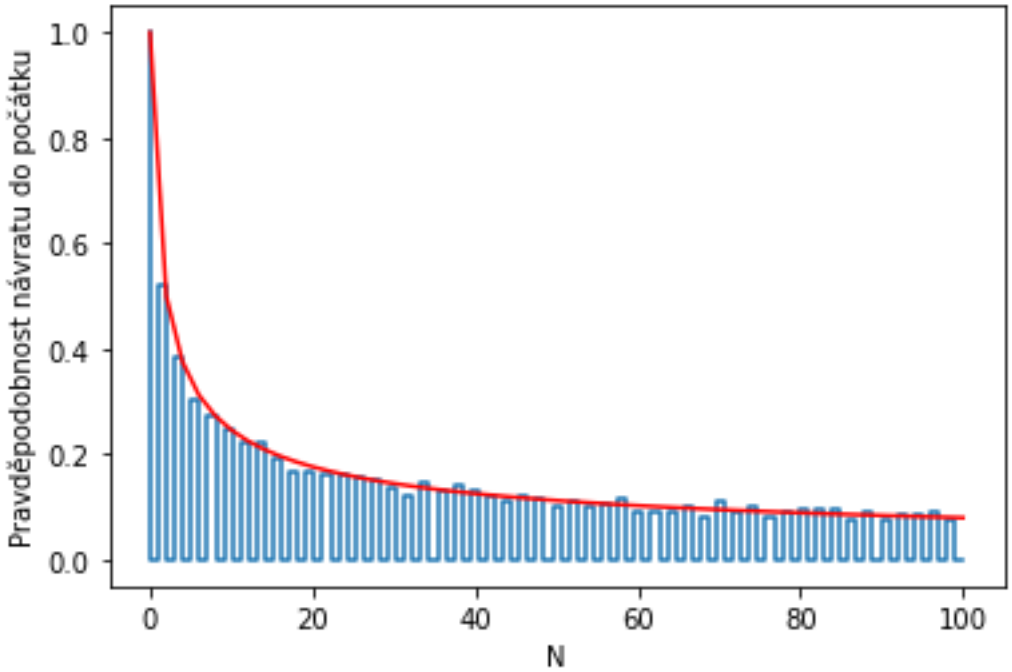
Náhodná procházka

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

random-walk.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import factorial
Nsim=1000 #pocet nahodnych prochazek, ktere budeme simulovat
N=100 #počet kroku budu postupne zvysovat od 0 do N
p=0.5 #pravdepodobnost ze pujdu vpravo
k_data=np.array(Nsim,dtype=int) #pocet kroku vpravo, binom. rozdeleni
x_N=np.linspace(0,N,N) #x osa pro vykresleni sqrt(N)
f=np.sqrt(x_N) #odmocninova zavislost
x_N_odd=np.arange(0,N+2,2,dtype=int) #pole sudych cisel N
P_teor=factorial(x_N_odd)/(factorial(x_N_odd/2))**2*(1/2)**x_N_odd # teoreticka P ze se
distance=np.array(Nsim) #vzdalenosti od pocatku
distance_mean=np.empty(N) #prumerna vzdalenost
P_zero=np.empty(N) #P ze se vratim do nuly

for i in range(0,N): #sumulujeme pocet kroku od 0 do N
    distance_zero_count=0 #pocet pripadu kdy jsem se vrati
    k_data=np.random.binomial(i,p,Nsim) #vygenerujeme Nsim
    distance=(k_data - (i-k_data))**2 #vzdalenost od pocat
    for k in range(0,Nsim):
        if distance[k]==0: distance_zero_count=distance_ze
    distance_mean[i]=np.sqrt(np.mean(distance)) #prumerna
    P_zero[i]=distance_zero_count/Nsim #P ze se vratim do

fig,ax=plt.subplots()   #graf zavislosti vzdalenosti od po
ax.set_xlabel("N")
ax.set_ylabel("vzdálenost od počátku")
ax.step(x_N,distance_mean)
ax.plot(x_N,f,c="red")

fig,ax=plt.subplots() #graf zavislosti P ze se vratim do p
ax.set_xlabel("N")
ax.set_ylabel("Pravděpodobnost návratu do počátku")
ax.step(x_N,P_zero)
ax.plot(x_N_odd,P_teor,c="red")
```