

Poissonovo rozdělení

Simulace experimentu Irvine-Michigan-Brookhaven v Pythonu

Possion-simulace.py

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

def Poisson(k,nu):
    return(nu**k*np.exp(-nu)/np.math.factorial(k))

vPoisson=np.vectorize(Poisson) #vektORIZACE funkce Poisson

N=2306 #pocet hodnot
nu=0.777 #ocekavana hodnota
data=np.random.poisson(nu,N) #simulace Poissonova rozdeleni

k=np.arange(0,9,1,dtype=int)
P=vPoisson(k,nu)

plt.hist(data,bins=k,density='True')
plt.plot([0,1],[P[0],P[0]],c='red')
plt.step(k+1,P,c='red')
plt.xlim(0,8)
```

Histogram

Zjistěte jak závisí chyba výšky i -tého binu histogramu na celkovém počtu naměřených hodnot N .

Práce s histogramy v Pythonu:

vytvoření histogramu

```
hist, bin_edges=np.histogram(data, bins=nbins, density='True')
```

↑
pole obsahující počty hodnot
v jednotlivých binech

↑
pole obsahující
hranice binů

↑
data, z
kterých se má
udělat
histogram

↑
počet binů

↑
normalizace
histogramu

Histogram

Zjistěte jak závisí chyba výšky i -tého binu histogramu na celkovém počtu naměřených hodnot N .

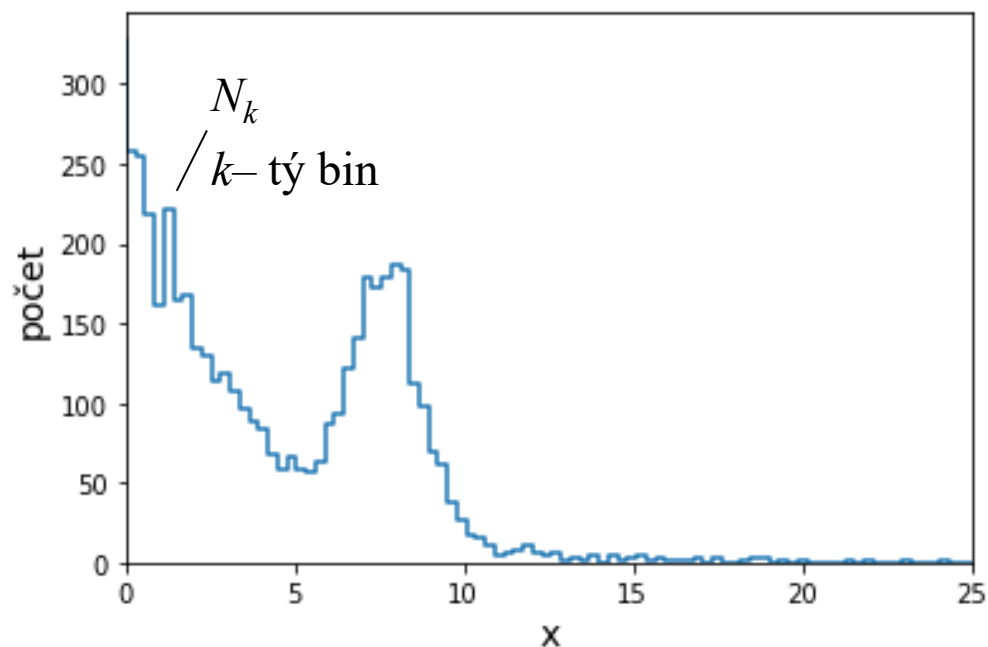
1. Vygeneruj N_{tot} dat \rightarrow naplň histogram \rightarrow zjisti počet případů N_k v k -tém binu
2. To opakuj N_{sim} – krát a spočítej aritmetický průměr hodnot N_k z každé simulace

$$\langle N_k \rangle = \frac{1}{N_{\text{tot}}} \sum_{i=1}^{N_{\text{sim}}} N_{ki}$$

3. Spočítej odhad rozptylu hodnot $N_{k,i}$ z jednotlivých simulací

$$\sigma_k^2 = \frac{1}{N_{\text{sim}} - 1} \sum_{i=1}^{N_{\text{sim}}} (N_{ki} - \langle N_k \rangle)^2$$

4. Opakuj 1.-3. pro různé hodnoty N_{tot}
a vynes závislost σ_k na N_{tot} .



Histogram

Program, v Pythonu

hist-bin-err.py

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

# simulace nah. promemne s rozdeleni exp+gaussian
def sim():
    branching_ratio=0.7
    r=np.random.random_sample()
    if r>branching_ratio:
        return np.random.normal(8,1)
    else:
        return np.random.exponential(3)

nbins=100 #pocet binu
ibin=10 #divame se na pocky hodnot v 10-tem binu
Nopakovani=100 #pocet opakovani pro kazde Ntot
Ntot=np.arange(100,5000,100) #ruzne pocky Ntot
hodnota=np.empty(Nopakovani) #pole hodnot v 10-tem binu
fraction=np.empty(np.size(Ntot)) #frakce hodnot pripadajici na 10-ty bin
std=np.empty(np.size(Ntot)) #pole standardnich odchylek
ik=0
for k in Ntot: #cyklu pro postupne narustajici celkove pocky dat Ntot
    print('počet dat = ',k)
    for j in range(Nopakovani): #100 opakovani por kazde Not
        data=np.empty(k)
        for i in range(k): #generovani histogramu
            data[i]=sim()
        hist,bin_edges=np.histogram(data,bins=nbins)
        hodnota[j]=hist[ibin]
    fraction[ik]=np.mean(hodnota)/k #jaka frakce hodnot podne do 10-teho binu
    std[ik]=np.std(hodnota) #standardni odchylka pro pocky v 10-binu
    ik+=1
fraction_mean=np.mean(fraction) #prumerna frakce pripadajici na 10-ty bin

fig,ax=plt.subplots()
plt.scatter(Ntot,std)
plt.plot(Ntot,np.sqrt(Ntot*fraction_mean),c='red')
ax.set_xlabel('$N_{tot}$',fontsize=14)
ax.set_ylabel('$\sigma$',fontsize=14)
```

Histogram

Zjistěte jak závisí chyba výšky i -tého binu histogramu na celkovém počtu naměřených hodnot N .

