

Наумов П. 3413

Д3: смоделировать вычисления 2.22, 2.27, 2.23, 2.25, 2.30

```
Ввод [4]: import numpy as np
from scipy.stats import binom, poisson, norm
from math import sqrt

np.set_printoptions(formatter = {'float_kind': '{:0.4f}'.format})
```

```
Ввод [6]: # 2.22 а)
# Пуассон
n = 3650
p = 1/365
lmb = n * p
m = 10
prob = poisson.pmf(k=m, mu = lmb)
print(f'{prob:.4f}')
```

0.1251

```
Ввод [15]: # 2.22 б)
# По Бернулли
s=0
for x in range(3,3651):
    s+=binom.pmf(x, 3650, 1/365)
print(f'{s:.4f}')
```

0.9973

```
Ввод [17]: # 2.27 а)
# Локальная теорема Муавра-Лапласа
n = 400
p = 0.5
q = 1 - p
m = 180

print(n * p * q, ">= 20")
x = (m - n * p) / sqrt(n * p * q)
prob = norm.pdf(x) / sqrt(n * p * q)
print(f'Вероятность равно (Муавра-Лаплас): {prob:.4f}')
```

100.0 >= 20

Вероятность равно (Муавра-Лаплас): 0.0054

Ввод [21]: `# 2.27 б)
Интегральная теорема Муавра-Лапласа
n = 400
p = 0.5
a = 180
b = 400`

```
mu = n*p  
sigma = np.sqrt(n * p * (1-p))  
x1 = (a - mu) / sigma  
x2 = (b - mu) / sigma  
prob = norm.cdf(x2) - norm.cdf(x1)  
print(f'Вероятность: {prob:.4f}')
```

Вероятность: 0.9772

Ввод [22]: `# 2.23 а)
Пуассон
n = 10000
p = 0.0001
lmb = n * p
print(lmb, "<= 10")
m = 5
prob = poisson.pmf(k=m, mu = lmb)
print(f'{prob:.4f}')`

1.0 <= 10
0.0031

Ввод [23]: `# 2.23 б)
По Бернулли
s=0
for x in range(0,3):
 s+=binom.pmf(x, 10000, 0.0001)
print(f'{s:.4f}')`

0.9197

Ввод [29]: `# 2.25 а)
ПО БЕРНУЛЛИ, т.к. n мало
p = 0.6
n = 10
m = 6
prob = binom.pmf(m, n, p)
print(f'{prob:.4f}')`

0.2508

```
Ввод [36]: # 2.25 б)
# Бернулли
n = 200
p = 0.6
lmb = n * p
print(lmb, "<= 10")
m = 120
prob = binom.pmf(m, n, p)
print(f'{prob:.4f}')
```

```
120.0 <= 10
0.0575
```

```
Ввод [5]: # 2.30
# Пуассон
p = 0.01
n = 800
m = 8
print(n*p, "<=10")
prob = poisson.pmf(k=m, mu=n*p)
print(f'{prob:.4f}')
```

```
8.0 <=10
0.1396
```