In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import bernoulli, binom, poisson, gamma, norm, uniform
```

Ριψη νομίσματος

$X \sim \mathrm{Be}(p)$

$p(k) = \begin{cases} p, & k = 1 \\ 1-p, & k = 0 \end{cases}$

$\mathbb{E}(X) = p, \quad \mathrm{Var}(X) = p(1-p)$

In [3]:
```
p_head = 0.1
distr = bernoulli(p_head)
```

In [4]:
```
N = 10000
data = distr.rvs(N)
```

In [5]:
```
data
```

Out[5]:
```
array([1, 0, 0, ..., 0, 0, 0])
```

In [6]:
```
np.mean(data)
```

Out[6]:
```
0.1033
```

Αριθμός κεφαλών σε n ανεξάρτητες ρίψεις νομίσματος

$X \sim \mathrm{Bin}(n, p)$

$p(k) = \binom{n}{k} p^k (1-p)^{n-k}, \; 0 \le k \le n$

$\mathbb{E}(X) = np, \quad \mathrm{Var}(X) = np(1-p)$

In [7]:
```
p_head = 0.1
N1 = 100
distr = binom(N1, p_head)
```

In [8]:
```
N2 = 1000
data = distr.rvs(N2)
```

In [9]:
```
data
```

Out[9]:
```
array([12,  6, 11,  9, 10,  6, 10,  7, 18, 12, 10,  8,  7, 13, 13,  7, 14,
        7, 11,  6,  9,  7,  9, 14,  9, 11,  7, 11,  8, 14, 11,  7,  8, 14,
       19, 16, 18,  6, 16,  5, 17,  5,  8, 14,  7, 13, 11, 12,  9, 11,  4,
       17, 15, 10, 11,  8,  8,  9, 12, 14,  2,  8, 16,  7,  8, 11,  9,  7,
       10, 11,  9, 11, 10,  8,  7, 12,  6, 11, 14, 10,  6,  9, 12, 10,  9,
        9,  8,  8, 17, 11, 10,  8, 12, 14, 11, 14, 13, 13, 11, 11,  8, 11,
        5,  6, 17, 10,  8,  8, 11, 13, 11,  9,  7,  9, 16,  5, 10,  8, 15,
       13, 14, 15, 11, 10,  6, 10, 13,  8, 12,  8,  9, 10, 11, 10,  7, 11,
        9,  6, 10,  6,  9, 15, 12, 10, 11, 12,  9, 12, 11, 13, 10,  9, 10,
        7, 14,  9, 12, 12, 13, 13, 11, 10,  7,  6, 10, 11, 11, 12,  4, 12,
        5, 15, 10, 15, 11, 13, 11, 14, 13, 12,  7, 13, 13,  8, 14, 10, 10,
       11,  5,  5,  6,  9,  6, 15,  9,  5,  7,  8,  7,  8, 10, 13,  8, 13,
       12, 11, 16, 13,  8, 14, 11, 10, 10,  9, 15, 11, 11, 11,  9, 11, 19,
       10,  7,  9, 12, 10, 10, 14, 12, 11, 14,  9, 10,  8, 12, 11, 11, 12,
       10, 14,  5,  9,  8,  8,  9, 15, 16, 16, 10, 12, 11, 12,  8,  8,  7,
       10, 10, 17, 11,  6, 15, 10, 13, 10, 11, 13, 11, 11, 14, 11,  9, 11,
       15, 11, 11,  6,  9, 13, 11, 14,  7, 14,  7,  6, 12,  6,  9, 11,  8,
        7, 11, 12,  9,  7, 11, 10,  9,  7, 10, 14, 14, 12,  9, 11,  8,  9,
       14, 15, 10,  9,  8,  6,  7,  6,  9, 10, 10, 13,  9, 10, 12,  8, 13,
       11, 16,  9, 10,  8,  8,  5,  8, 15,  9,  4, 12, 13, 10, 10,  5, 14,
       14, 10,  6,  8,  8, 16,  8, 10, 11, 11,  9,  9, 11, 10,  9, 13, 15,
        8,  9, 12,  8, 12, 10,  7,  6,  8, 16,  7, 15,  7,  9, 12,  8,  6,
       10,  8, 13,  9,  8,  9,  9, 10, 10,  7,  9, 13,  3,  4, 10, 12,  9,
       12, 10,  8,  3, 10,  9, 17, 13,  8,  9, 10, 10, 12,  4, 14, 12,  8,
        6,  8,  9, 11, 11, 10,  7,  6,  4, 10, 10,  7,  4, 10, 13,  9,  9,
       13, 11,  9, 11,  8,  6, 16, 11,  5, 10,  6,  5, 17,  8,  9, 13,  7,
        9,  8, 10, 11, 11,  8,  7,  9, 10, 10,  6, 12, 14,  9, 12,  6,  5,
        6, 11,  9, 10,  6, 11, 13, 16,  6,  8, 13, 13, 11,  6,  5, 11,  9,
       11, 11, 10,  7,  9, 15, 14,  4, 10,  5, 13,  6,  9, 10,  8,  5,  8,
       15,  7, 10,  9, 12,  9,  5,  8, 14, 15, 10,  3,  8,  7, 11,  7, 10,
        5, 10,  6,  9, 17, 14,  7, 17,  9,  7, 11,  9, 15, 10, 11, 10, 16,
        6,  9,  8,  9,  6, 10, 12, 12,  9,  6, 10, 11,  7,  9, 10,  6,  8,
        3, 14,  8,  5, 15,  9, 10, 12, 12,  7, 11, 11,  8,  8,  4, 13, 11,
        8,  7,  9,  8, 11,  8, 11, 13, 12, 13, 16,  9, 10,  4, 10,  6,  7,
       11,  6,  6, 10, 14, 11, 15,  5,  7, 11, 11,  8, 10,  6, 14, 10, 10,
       13, 12, 12, 10,  9, 15,  9, 10, 15, 13, 11, 14, 10, 11, 11,  8,  7,
       15,  9,  7, 15,  8, 10,  8,  9,  4,  9,  4, 12, 10, 16, 10, 10, 13,
        7, 13,  3, 12,  7,  5,  8, 15, 14, 11,  6,  5,  8, 11, 10, 10, 11,
       11, 15, 16,  6, 13, 11, 14, 10, 13,  9,  5, 13, 11, 10,  8,  8, 10,
        9,  8,  8, 17, 10, 10, 10, 10,  7,  9,  8,  7,  6, 16, 13,  9, 12,
       14, 14, 10,  6, 10,  3,  9, 11,  9, 10,  7,  7, 10,  9,  8,  8,  9,
       14,  7, 10,  7, 12, 14,  9,  7,  7, 12,  7,  7, 10,  5, 13, 11,  6,
        9, 11, 13, 13,  5,  8,  8, 14,  9, 13, 11, 10, 10, 10,  8, 15,  9,
       11,  8, 10, 12,  9, 16, 10,  8, 11,  9, 10, 14, 11, 11,  8, 12, 11,
       11,  9, 11,  6,  8, 11,  5,  8, 10, 10,  7, 12,  8,  8,  8, 13, 12,
       10, 11, 10,  9, 10, 15, 11, 10,  4,  8, 11, 11,  9,  9,  8, 12, 11,
       10, 15, 13, 10, 13, 12,  9, 12, 12, 15, 12,  9, 14, 12, 12,  5, 11,
        8, 15,  7,  5, 10, 10, 12, 11, 14, 12,  9,  8, 11, 10,  9, 10,  6,
       14,  6, 11, 13,  9,  9,  7, 16, 15, 13, 12,  5, 11, 10, 10,  8, 16,
       10,  8, 10, 12,  8, 12,  9, 10, 13, 11, 14,  4,  7, 12, 15, 19,  7,
       10, 10,  8,  8, 13,  9, 10, 10, 10, 12,  2,  8, 20,  4, 10,  9,  6,
        9,  8,  6,  6, 10,  9, 13, 11,  4,  9,  8, 13, 13,  9, 10, 15,  9,
       11,  7, 12,  7, 13, 13, 11, 11, 16, 12, 12, 14, 10,  7,  6,  8, 11,
       12,  6, 11, 11,  9, 17, 15,  5,  7,  7, 10, 13, 14,  9, 12,  7,  8,
        6,  8, 12,  9,  7,  7, 14, 12, 11, 12, 13, 11,  8, 16,  6, 15,  6,
        7, 11, 15, 17,  8,  8,  5, 10,  6,  9,  5, 16, 12,  9,  4,  7,  8,
       15, 10,  7, 10,  6, 10,  8,  8, 13, 11,  6,  9, 13,  7, 13,  8,  9,
        9,  3, 10,  7,  6, 11, 11, 12,  7,  9,  9,  9, 12, 13, 10,  7, 11,
        6,  8, 10,  7, 12, 10,  9,  9,  4,  9, 13, 13, 11,  8])
```

Ο αριθμός των ψαριών που αλιεύει ένας ψαράς σε καθορισμένο χρονικό παράθυρο

$X \sim \mathrm{Po}(\lambda),\ \lambda > 0$

$p(k) = \frac{\lambda^k e^{-\lambda}}{k!},\ k \geq 0$

$\mathbb{E}(X) = \lambda, \quad \mathrm{Var}(X) = \lambda$

In [10]:
```
l = 5
distr = poisson(l)
```

In [19]:
```
N = 100
data = distr.rvs(N)
```

In [20]:
```
data
```

Out[20]:
```
array([ 3,  6,  6,  4,  6,  4, 10,  2,  4,  8,  5,  3,  5,  6,  4,  5,  3,
        6,  5,  4,  2,  2,  5,  7,  9,  5,  3,  4,  8,  0,  5,  9,  6,  4,
        3,  6,  3,  9,  4,  5,  2,  6,  6,  3,  9,  5,  7,  6,  7,  8,  8,
        3,  4,  2,  5,  9,  3,  4,  3,  3,  7,  4,  5,  4,  7,  9,  3,  4,
        5,  5,  4,  2,  6,  7,  5,  3,  3,  6,  5,  4,  1,  1,  3,  4,  6,
        7,  4,  3,  7,  4,  4, 10,  1,  7,  4,  3,  2,  7,  5,  2])
```

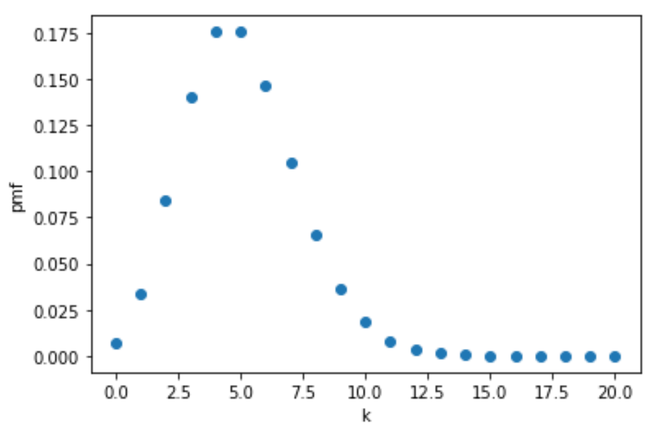
In [27]:
```
x = np.arange(0, 21)
poisson_pmf = distr.pmf(x)
```

In [28]:
```
poisson_pmf
```

Out[28]:
```
array([6.73794700e-03, 3.36897350e-02, 8.42243375e-02, 1.40373896e-01,
       1.75467370e-01, 1.75467370e-01, 1.46222808e-01, 1.04444863e-01,
       6.52780393e-02, 3.62655774e-02, 1.81327887e-02, 8.24217669e-03,
       3.43424029e-03, 1.32086165e-03, 4.71736303e-04, 1.57245434e-04,
       4.91391982e-05, 1.44527054e-05, 4.01464038e-06, 1.05648431e-06,
       2.64121077e-07])
```

In [29]:
```
plt.plot(x, poisson_pmf, 'o')
plt.xlabel('k')
plt.ylabel('pmf')
```

Out[29]:
```
Text(0, 0.5, 'pmf')
```



Χρόνος που απαιτείται για αλίευση ενός αριθμού ψαριών (σύνδεση με Poisson)

$X \sim \mathrm{Gamma}(\alpha, \beta),\ \alpha > 0,\ \beta > 0$

$f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad ό\pi o\upsilon\ \Gamma(\alpha) = \int_0^\infty s^{\alpha-1} e^{-s} ds$

$\mathbb{E}(X) = \frac{\alpha}{\beta}, \quad \mathrm{Var}(X) = \frac{\alpha}{\beta^2}$

In [51]:
```
alpha = 6
beta = 1
distr = gamma(alpha, scale = 1/beta)
```

In [52]:
```
N = 100
data = distr.rvs(N)
```

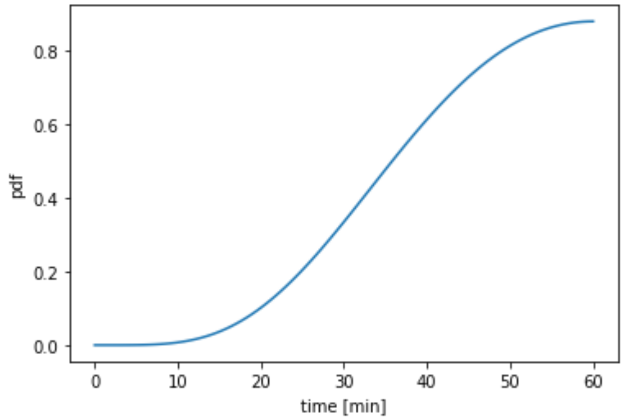
In [53]:
```
data * 60
```

Out[53]:
```
array([106.24219151,  78.38586143,  71.04098566,  71.89749725,
        51.95822184,  47.09949596,  43.83321777, 114.16052561,
        87.37786525,  51.5162145 ,  82.6676022 ,  43.92316556,
        94.61999174,  82.91345134,  84.62535054, 156.65728283,
        56.88995217,  57.11375597,  75.29542697, 121.56766121,
        48.22776478,  96.43679937,  55.74701464,  79.1836195 ,
        92.4817812 ,  75.18548132,  54.42430958,  92.61829968,
        55.10868223,  68.07347411,  96.54973876, 119.38471271,
        79.73132148, 107.236799  ,  54.13687679, 102.3564341 ,
        72.51460623, 102.30357145,  71.83818485,  25.66377854,
        55.45551512,  96.20761942,  55.18339378,  54.12138202,
       164.80506395,  62.14342318,  58.94759048,  49.40354928,
       127.32188751,  35.03001172,  15.25423202,  39.38784074,
       104.44626225,  64.13803508,  76.50173289,  99.95880384,
        88.85302155,  18.25461349,  35.79585039,  83.33865076,
        93.83282495,  39.19623921,  72.29911486,  43.72540437,
        84.70759779,  90.57009381, 133.83162011,  81.25104475,
        78.12646185,  88.36956845,  43.15632567,  43.28579476,
       131.74012422,  91.89598513,  71.26622924,  50.94024847,
        69.94695878,  92.71927679,  66.98408495,  78.05723907,
        96.59680261,  60.51882231,  67.18830372,  59.13925394,
        85.50161707,  72.26549557,  39.08471218,  80.62238891,
        89.60331684,  69.43417453, 176.91039046,  62.4235459 ,
        43.27190697,  48.97297529,  53.09153921,  82.52473977,
        61.39081515,  61.19098769,  41.59295438,  64.78897725])
```

In [56]:
```
x = np.linspace(0,1,100)
gamma_pdf = distr.pdf(x)
```

In [57]:
```
plt.plot(x * 60, gamma_pdf)
plt.xlabel('time [min]')
plt.ylabel('pdf')
```

Out[57]: Text(0, 0.5, 'pdf')



Ύψος των γυναικών μιας χώρας

$X \sim \mathcal{N}(\mu, \sigma^2),\ \mu \in \mathbb{R},\ \sigma > 0$

$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

$\mathbb{E}(X) = \mu, \quad \mathrm{Var}(X) = \sigma^2$

In [58]:
```
mu = 165.0
sigma = 6.5
distr = norm(mu,sigma)
```

In [62]:
```
data = distr.rvs(10)
```

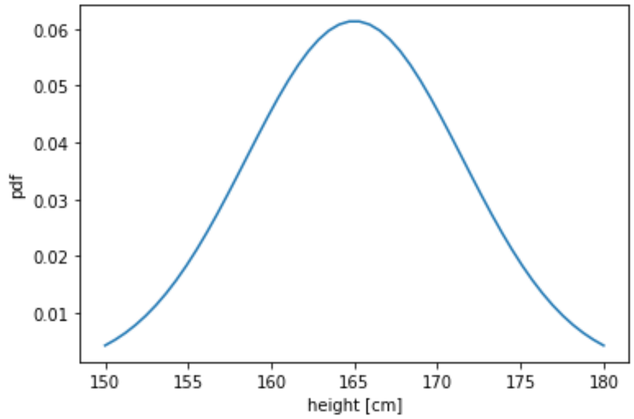
In [63]:
```
data
```

Out[63]:
```
array([167.20388516, 161.16739173, 159.17719756, 168.56413285,
       165.64133845, 166.28167686, 171.88367715, 158.35333146,
       159.44093703, 173.17192773])
```

In [59]:
```
x = np.linspace(150,180)
norm_pdf = distr.pdf(x)
```

In [60]:
```
plt.plot(x, norm_pdf)
plt.xlabel('height [cm]')
plt.ylabel('pdf')
```

Out[60]: Text(0, 0.5, 'pdf')



Ομοιόμορφη κατανομή

$X \sim U[a, b], \;\; -\infty < a < b < \infty$

$f(x) = \frac{1}{b-a}, \;\; -\infty < a < b < \infty$

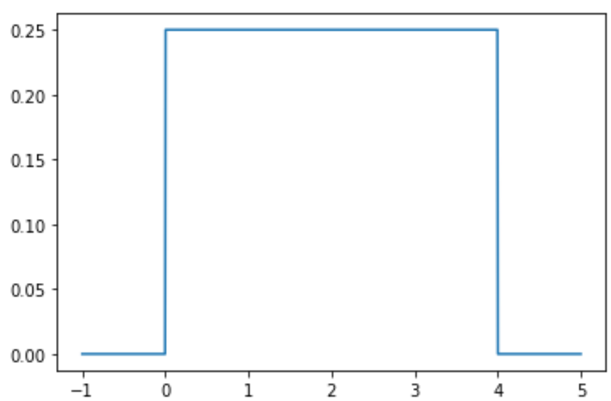
$\mathbb{E}(X) = \frac{a+b}{2}, \quad \mathrm{Var}(X) = \frac{(b-a)^2}{12}$

In [64]:
```
a = 0
b = 4
distr = uniform(a, b - a)
```

In [65]:
```
x = np.linspace(-1,5,1000)
uniform_pdf = distr.pdf(x)
```

In [66]:
```
plt.plot(x, uniform_pdf)
```

Out[66]: [<matplotlib.lines.Line2D at 0x7f76c2604290>]



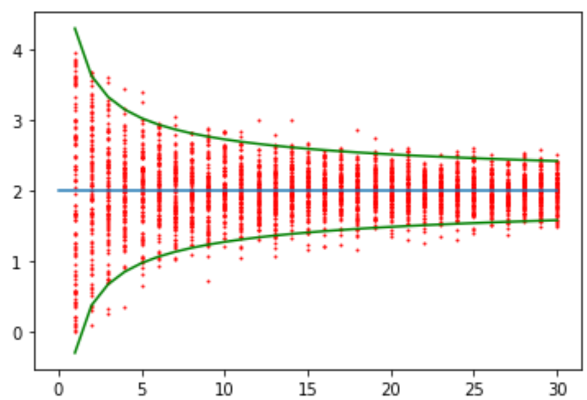
Νόμος των μεγάλων αριθμών

$$X_1, X_2, \dots \; i.i.d, \quad \mathbb{E}(X_1) = \mu, \; \mathrm{Var}(X_1) = \sigma^2$$
$$\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$
$$\mathbb{E}(\bar{X}_n) = \mu, \; \mathrm{Var}(\bar{X}_n) = \sigma^2/n$$
$$\bar{X}_n \xrightarrow{p} \mu$$

In [82]:
```
def var_bar_X(sigma2, n):
    return sigma2 / n
```

In [84]:
```
np.random.seed(1)
N = 30
Q = 100
epsilon = 0.01
for n in range(1,N+1):
    for q in range(Q):
        data = distr.rvs(n)
        # print(data, np.mean(data))
        plt.plot(n, np.mean(data), 'o', ms=1, color = 'r')
plt.plot((0,N), (0.5 * (a + b), 0.5 * (a + b)))
plt.plot(np.linspace(1,N,N), 2 + 2 * np.sqrt(var_bar_X((b-a)**2/12, np.linspace
(1,N,N))), color = 'g')
plt.plot(np.linspace(1,N,N), 2 - 2 * np.sqrt(var_bar_X((b-a)**2/12, np.linspace
(1,N,N))), color = 'g')
```

Out[84]: [<matplotlib.lines.Line2D at 0x7f76b6793b10>]



In [ ]: