Hints for the solution of the 4th assignment

A. Discrete random variables

During the writing and testing the code you may try the next or similar interactive sections (the first line shows, how to read and run your code into an interactive section):

```
>>> exec(open('5YourName.py').read())

>>> y = Drv([25, 33, 40, 50], [1/4]*4 )
>>> y.xk
[25, 33, 40, 50]
>>> y.pk
[0.25, 0.25, 0.25, 0.25]
>>> y
Discrete random variable: (25,0.25) (33,0.25) (40,0.25) (50,0.25)
>>> y.e()
37.0
>>> x = y.reweight()
>>> x
Discrete random variable: (25,0.16891891891891891) (33,0.22297297297297297)
(40,0.2702702702702703) (50,0.33783783783783783)
>>> x.cdf(42)
0.6621621621621622
>>> x.cdf(52)
1.0

>>> b = Binomial(11, .5)
>>> b.cdf(6)
0.5
>>> b
Binomial random variable: (0,0.00048828125) (1,0.00537109375) (2,0.02685546875)
(3,0.08056640625) (4,0.1611328125) (5,0.2255859375) (6,0.2255859375) (7,0.1611328125)
(8,0.08056640625) (9,0.02685546875) ...
>>> b.cdf(0)
0
>>> b.cdf(1)
0.00048828125
>>> b.pdf(0)
0.00048828125
>>> br = b.reweight()
>>> br
Discrete random variable: (0,0.0) (1,0.0009765625) (2,0.009765625) (3,0.0439453125)
(4,0.1171875) (5,0.205078125) (6,0.24609375) (7,0.205078125) (8,0.1171875) (9,0.0439453125) ...

>>> z = Uniform(3)
>>> z
Uniform random variable: (1,0.3333333333333333) (2,0.3333333333333333) (3,0.3333333333333333)
>>> z.e()
2.0
>>> z.pdf(2)
0.3333333333333333
>>> z.cdf(3)
0.6666666666666666
>>> z.cdf(4)
1.0
```

B. The party starts

- You may use the next code for drawing graphs (other good solutions are possible!):

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

# t will be a list with elements 0, .02, .04,..., 1
t = np.linspace(0, 1, 51)
plt.plot(t, dfunc(t), color = "red")
```

- Here the first argument of the function `plot` is the list of the values of the domain, and the second argument contains the values of the function at these points, where `dfunc` have to be defined before. Drawing a histogram means drawing columns over intervals the height of which are proportional to the number of simulated values in the interval. There is a simple Python function to draw this histogram, where `experiments` is the list of the values of X, `bins` is a list of values in the domain, `density`means the total area of the columns:

```
plt.hist(experiments, bins=t, density=1)
```

- To draw points, you need to specify the style as the third argument of the function `plt.plot`, in this task it can be "r.", which means that the specified points will not be joined together, and the color of the points will be green (r = red).