## Problema 1

Timpul de întârziere în minute al unui student la un examen este o variabilă aleatoare $T$ cu distribuția exponențială cu media $2$ minute, adică $Exp(1/2)$. Indepentă de $T$, nota obținută de student la examen are distribuția uniformă discretă, adică $Unid(10)$. Profesorul scade un punct din nota studentului pentru fiecare minut întreg din timpul de întârziere al studentului (dacă se obține o notă mai mică decât $1$, atunci profesorul consideră nota finală $1$). Fie $N$ nota finală a studentului.

**a)** Simulați $10000$ de valori pentru $T$, apoi afișați o histogramă a frecvențelor relative cu $10$ bare pe intervalul $[0, 10]$ și graficul funcției de densitate pe acest interval.

**b)** Estimați $P(N \geq 5)$, apoi afișați probabilitatea teoretică.

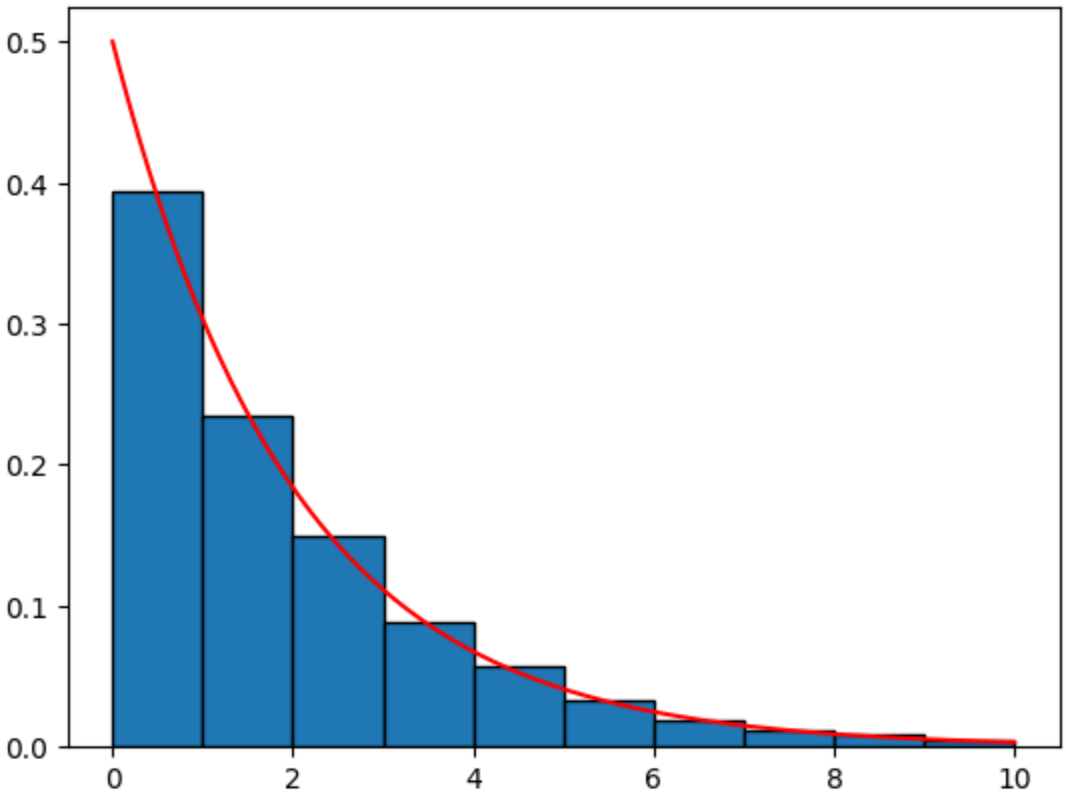
In [1]:
```
from scipy.stats import expon, randint
from matplotlib.pyplot import hist, plot
from numpy import linspace, mean, floor

timpi = expon.rvs(scale=2,size=10000)

hist(timpi,10,range=(0,10),density=True,edgecolor='k')
x = linspace(0,10,1000)
plot(x,expon.pdf(x,scale=2),'-r')

print("P(N≥5) ≈ ", mean([randint.rvs(1,11)-floor(t)>=5 for t in timpi]))
print("P(N≥5) = ", sum([randint.pmf(k,1,11)*expon.cdf(k-4,scale=2) for k in range(5,11)]))
```

```
P(N≥5) ≈  0.4559
P(N≥5) =  0.45352523887391194
```




In [ ]:

In [ ]:

In [ ]:

## Problema 2

Dintr-o populație se alege aleator, cu returnare, câte o persoană până când se găsește o persoană cu înălțimea mai mare decât $1,90$ m. Fie $X$ numărul de persoane alese. Știind că înălțimea unei persoane alese aleator urmează distribuția normală cu media $1,65$ m și deviația standard $0,20$ m,

**a)** generați $10000$ de valori pentru $X$, apoi afișați o histogramă a frecvențelor relative pentru valorile: $1, 2, \ldots, 10$.

**b)** Estimați $P(X > 10)$, apoi afișați probabilitatea teoretică.

In [2]:
```
from scipy.stats import norm, geom
from matplotlib.pyplot import hist, xticks
from numpy import mean

prob_succes = 1 - norm.cdf(1.90,loc=1.65,scale=0.2)

x = geom.rvs(prob_succes,size=10000)
bin_edges = [k+0.5 for k in range(0,11)]
hist(x,bin_edges,density=True,edgecolor='k')
xticks(range(1,11))
print("P(X>10) ≈ ", mean(x>10))
print("P(X>10) = ", 1-geom.cdf(10,prob_succes))
```

```
P(X>10) ≈  0.3258
P(X>10) =  0.32739814342327334
```