STAT 753: Stochastic Models and Simulations

Jaleesa Houle
University of Nevada, Reno - Spring 2024

Homework 1

1. If a fair coin is flipped, find the probability that the first Head appears on the third toss.

Solution

The sample space for a coin flipped 3 times is

$$\{HHH, TTT, HHT, THH, HTH, THT, TTH, TTH\}. \tag{1}$$

The probability of an event occurring is:

$$P(E) = \frac{\#\text{ favorable outcomes}}{\#\text{ total outcomes}}. \tag{2}$$

In this case, there is one favorable outcome out of eight possible outcomes, so the probability is $\frac{1}{8}$ or 12.5%. Alternatively, we can see that each coin toss is an independent event, with $P(H) = P(T) = \frac{1}{2}$. Therefore, the probability of flipping heads on the third toss is equal to

$$P(T) \cdot P(T) \cdot P(H) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}. \tag{3}$$

Code

```
np.random.seed(10)
toss_trials = []

p=0.5 #probability for a fair coin
N=1000 #desired simulations

#define Heads=1, Tails=0
for i in range(N):
    toss_trials.append(np.random.binomial(1,p, size=3))

desired_outcome=[0,0,1]
total_TTH_events = [toss_trials[i] for i in range(N) if all(toss_trials[i]
↪   == desired_outcome)== True]
prob1 = (len(total_TTH_events)/N)
print('Probability of event {T,T,H} based on 1000 simulations: %s' % prob1)
print('Theoretical probability of event {T,T,H}: 0.125')
print('Percent error: %s%%' % np.round(((prob1 - 0.125)/0.125)*100,3))

>> Probability of event {T,T,H} based on 1000 simulations: 0.129
>> Theoretical probability of event {T,T,H}: 0.125
>> Percent error: 3.2%
```

2. Assume 100 houses are independently on fire, each with probability 2.5%. Find the probability that there are exactly 2 fires, using: (a) the binomial distribution; (b) the Poisson approximation.

Solution

(a) The probability that exactly 2 houses are on fire can be found by simply plugging in the known values to the binomial pdf:

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k} = \binom{100}{2} 0.025^2 (1-0.025)^{100-2} \tag{4}$$

The resulting probability is approximately 0.25878.

(b) The Poisson pdf is defined as:

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{5}$$

We can compute the probability of exactly 2 houses on fire by noting that k=2 and $E[\lambda] = n \cdot p = 100 \cdot 0.025 = 2.5$. Therefore, the probability is calculated as:

$$\frac{\lambda^2 e^{-\lambda}}{2!} = \frac{2.5^2 e^{-2.5}}{2} = 0.2565. \tag{6}$$

Code

Part (a)

```
np.random.seed(100)
N=1000 #desired simulations
n_houses=100
p=0.025

binomial_sample = np.random.binomial(n_houses, p, size=N)
prob2a_sim = len([binomial_sample[k] for k in range(N) if binomial_sample[k]==2])/N
prob2a_theory = binom.pmf(k=2, n=100, p=0.025)

print('Probability of exactly 2 fires based on 1000 simulations with binomial dist: %s' % prob2a_sim)
print('Theoretical probability of exactly 2 fires with binomial dist: %s' % np.round(prob2a_theory,4))
print('Percent error: %s%%' % np.abs(np.round(((prob2a_sim - prob2a_theory)/prob2a_theory)*100,3)))

>> Probability of exactly 2 fires based on 1000 simulations with binomial dist: 0.263
>> Theoretical probability of exactly 2 fires with binomial dist: 0.2588
>> Percent error: 1.629%

```

Part (b)

```
np.random.seed(1000)
N=1000 #desired simulations
n_houses=100
p=0.025
poisson_sample = np.random.poisson(lam=n_houses*p,size=N)

prob2b_sim = len([poisson_sample[k] for k in range(N) if poisson_sample[k]==2])/N
prob2b_theory = poisson.pmf(k=2, mu=100*0.025)

print('Probability of exactly 2 fires based on 1000 simulations with poisson dist: %s' % prob2b_sim)
print('Theoretical probability of exactly 2 fires with poisson dist: %s' % np.round(prob2b_theory,4))
print('Percent error: %s%%' % np.round(((prob2b_sim - prob2b_theory)/prob2b_theory)*100,3))

>> Probability of exactly 2 fires based on 1000 simulations with poisson dist: 0.273
>> Theoretical probability of exactly 2 fires with poisson dist: 0.2565
>> Percent error: 6.426%
```

3. Historical annual returns of S&P 500 have mean 0.067 and standard deviation 0.165. Assuming they are independent, find the probability that the return is positive if the distribution is: (a) Gaussian; (b) Laplace.

Solution

(a) For a Gaussian distribution $X \sim \mathcal{N}(0.067, (0.165)^2)$, the probability that an annual return is positive can be written as $P(0 < x < \infty) = 1 - F_x(0)$, where

$$F_x(0) = \Phi(\frac{x-\mu}{\sigma}) = \Phi(\frac{0-0.067}{0.165}) = \Phi(-0.4067). \tag{7}$$

By symmetry, $\Phi(-z) = 1 - \Phi(z)$. The probability can then be written as

$$P(0 < x < \infty) = 1 - (1 - \Phi(0.4067)) = \Phi(0.4067). \tag{8}$$

Using a standard normal distribution chart, the approximate probability is 0.6576.

(b) The Laplace pdf is defined as

$$f(x) = \frac{1}{2\lambda} e^{\frac{-|x-\mu|}{\lambda}} \tag{9}$$

where $\mu = 0.067$ and $\lambda = 0.165$. To find $P(0 < x < \infty)$, we need to integrate $f(x)$ over the desired bounds. Doing so yields

$$F(x > 0) = 1 - \frac{1}{2} e^{-\frac{|x-\mu|}{\lambda}} = 1 - \frac{1}{2} e^{-\frac{|0-0.067|}{0.165}} \tag{10}$$

The resulting probability is approximately 0.6669.

Code

Part (a)

```
np.random.seed(1000)
mu = 0.067
sigma= 0.165
N =1000
gaussian_sample = np.random.normal(mu, sigma, N)

prob3a_sim = len([gaussian_sample[k] for k in range(N) if gaussian_sample[k]>0])/N
prob3a_theory = 1 - norm.cdf(0, mu, sigma)

print('Probability of S&P 500 positive return with gaussian dist: %s' % prob3a_sim)
print('Theoretical probability of S&P 500 positive return with gaussian dist: %s' % np.round(prob3a_theory,4))
print('Percent error: %s%%' % np.round(((prob3a_sim - prob3a_theory)/prob3a_theory)*100,3))
```

```

>> Probability of S&P 500 positive return with gaussian dist: 0.662
>> Theoretical probability of S&P 500 positive return with gaussian dist: 0.6577
>> Percent error: 0.661%
```

Part (b)

```
np.random.seed(1000)
mu = 0.067
sigma= 0.165
N =1000
laplace_sample = np.random.laplace(mu, sigma, N)

prob3b_sim = len([laplace_sample[k] for k in range(N) if laplace_sample[k]>0])/N
prob3b_theory = 1 - laplace.cdf(0, mu, sigma)

print('Probability of S&P 500 positive return with laplace dist: %s' % prob3b_sim)
print('Theoretical probability of S&P 500 positive return with laplace dist: %s' % np.round(prob3b_theory,4))
print('Percent error: %s%%' % np.abs(np.round(((prob3b_sim - prob3b_theory)/prob3b_theory)*100,3)))

>> Probability of S&P 500 positive return with laplace dist: 0.652
>> Theoretical probability of S&P 500 positive return with laplace dist: 0.6669
>> Percent error: 2.229%
```