New York University Shanghai

Problem Set 13

Solutions

BUSF-SHU 210: Business Analytics (Spring 2019)

1. Application of For Loop in Python

The loop structure is very important in Python and other programing languages. In this problem, you are going to use `for` loop, which we learned in the lecture of Session 13, to solve two simple mathematical problems.

(a) A sequence $\{A_n : n \geq 1\}$ satisfies that $A_1 = 1$, $A_2 = 1$, and $A_{n+2} = A_{n+1} + A_n + n$ for $n \geq 1$. Use Python to find the value of A_{100}.

We use loop to calculate that $A_{100} = 1,281,597,540,372,340,801,536$.

(b) We define $\binom{n}{m}$ as the coefficient of binomial expansion for $(1 + x)^n$ $(m = 0, 1, 2, \cdots, n)$. Hence, $\binom{n}{m}$ is the coefficient of x^m in $(1 + x)^n$, where $m = 0, 1, 2, \cdots, n$. We know from the property of binomial coefficients that $\binom{n}{m} + \binom{n}{m+1} = \binom{n+1}{m+1}$ for all n and all $0 \leq m \leq n-1$. Please use R to calculate $\binom{100}{50}$. (*Hint:* You may need to use a matrix/array to store your calculations of $\binom{n}{m}$.)

$\binom{100}{50} = 100,891,344,545,564,237,256,087,044,096$.

2. Call Option Simulation

A European call option gives its owner the right to buy a particular stock at a given price (the strike price) on a specific date in the future (the expiration date). The movement of stock prices can be modeled using the following Geometric Brownian Motion:

$$S_{t+\Delta} = S_t(1 + \mu\Delta + z\sigma\sqrt{\Delta})$$

where

- S_t = price of the stock in time t
- $\mu = v + 0.5\sigma^2$
- v = the stock's expected annual growth rate
- σ = the standard deviation of the stock's annual growth
- Δ = time period interval, expressed in years
- $S_{t+\Delta}$ = price of the stock in time $t + \Delta$
- z = a standard normal random variable (with mean 0 and standard deviation 1)

Consider a stock with initial price $S_0 = \$80$, an expected annual growth rate $v = 0.15$, and a standard deviation $\sigma = 0.25$. Please briefly answer the following questions:

(a) Build a simulation model to simulate this stock's price behavior for the next 13 weeks? Plot simulated stock price for one trial. (*Hint*: The time of one week is 1/52)

The simulation model recursively computes S_t. The plot for a simulated trial of the stock price is given in 1.

(b) You are interested in purchasing a European call option written on this stock with a strike price \$75 and an expiration data at week 13. What is the expected profit you can earn if you own this option? Base you answer on the simulation model built in part (a).

First, we need to simulate the stock price in week 13, P_{13}. The profit from the European call option is $\max(P_{13} - 75, 0)$. Thus, the expected profit of the option is $\mathbb{E}\max(P_{13} - 75, 0)$. To calculate this, we simulate 10,000 P_{13}'s, obtain the respective profit for each simulated stock price, and take the average of all the simulated profits to get the expected profit. Using Python to perform the simulation, we have the expected profit is 9.79.

(c) If you own the option specified in part (b), what is the probability that you will earn a positive profit? Base you answer on the simulation model built in part (a).

First, we need to simulate the stock price in week 13, P_{13}. You can earn a positive if and only if $P_{13} > 75$. Thus, the probability of making a positive profit is $\mathbb{P}(P_{13} > 75)$. To calculate this, we simulate 10,000 P_13's, assign $Y = 1$ if $P_{13} > 75$ and $Y = 0$ if $P_{13} \leq 75$, and take the average of Y to get $\mathbb{P}(P_{13} > 75)$. Using Python to perform the simulation, we have the probability of positive profit is 0.799.

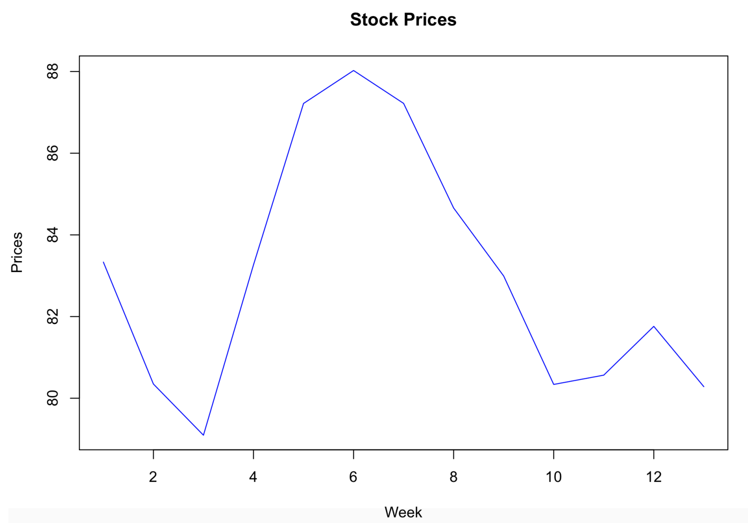

Figure 1: Simulated Stock Price

(d) Another type of option is the Asian call option, whose price depends on the average price until the expiration date. Specifically, suppose the Asian call option has a strike price of \$75 and an expiration date at week 13, so the payoff of this option is

$$\max\left(\frac{P_1 + P_2 + \cdots + P_{13}}{13} - 75, 0\right)$$

where P_i is the price of the stock at week i. If you own this option, what is the expected profit you can earn? What is the probability that you can earn a positive profit with this option?

First, we need to simulate the stock prices for weeks 1-13, $P_1, P_2, \cdots, P_{13}$. The profit from the Asian call option is $\max\left(\frac{P_1+P_2+\cdots+P_{13}}{13} - 75, 0\right)$ and you can make a positive profit if $\frac{P_1+P_2+\cdots+P_{13}}{13} > 75$. Thus, the expected profit of the Asian option is $\mathbb{E}\max\left(\frac{P_1+P_2+\cdots+P_{13}}{13} - 75, 0\right)$, and the probability of making a positive profit is $\mathbb{P}\left(\frac{P_1+P_2+\cdots+P_{13}}{13} > 75\right)$. We follow the same method as in parts (b) and (c) to obtain these quantities by simulation in Python. We have that the expected of Asian option is 7.08, and the probability of positive profit is 0.879.

Hint: To estimate the probability of an event to happen, you may simulate a large number of trials and count the proportion of trials in which the event happens.

3. Monte-Carlo Simulation for Multi-Armed Bandit

Use monte-carlo simulation to find out the following descriptive statistics of the regret for A/B testing, simple greedy, uniform exploration, and upper-confidence bound. We set the same parameters as the example discussed in Session 13's lecture. You can choose your own sample size.

- Expected total regret $\mathbb{E}[R(T)]$.
- Maximum total regret (*Hint*: You may use the function `numpy.maximum()`, `https://docs.scipy.org/doc/numpy/reference/generated/numpy.maximum.html`).
- Minimum total regret (*Hint*: You may use the function `numpy.minimum()` `https://docs.scipy.org/doc/numpy/reference/generated/numpy.minimum.html`).
- Mediam total regret (*Hint*: You may use the function `numpy.medium()` `https://docs.scipy.org/doc/numpy/reference/generated/numpy.medium.html`).
- 75% quantitle of regret (*Hint*: You may use the function `numpy.quantile()` `https://docs.scipy.org/doc/numpy/reference/generated/numpy.quantile.html`).
- 25% quantile of regret.

Also draw the histogram of regret for each algorithm discussed in Session 13.

Please refer to the Python code.