## Monte Carlo Simulation

09/04/2025

The simulation of covered call trading requires historical price data. For the SPY ETF, the available data are from the past 20 years for a sample size of approximately 5000. Statistically, this may not be sufficient to allow one to draw sound conclusions. Since the stock returns (not prices) largely follow a simple normal distribution, one can generate a large number of samples using the estimated mean and standard deviation from the historical stock returns. For example, the daily returns of SPY have $\mu$ =0.0004472206 and $\sigma$=0.01210494, whereas the daily returns of QQQ have $\mu$=0.0005332306 and $\sigma$ =0.01710912.

### Simulation Procedure

The R script for the simulation is listed below. *retn()* is a function for generating stock returns from the stock prices. Conversely, *prz()* is a function to generate the daily prices from the simulated daily returns. *c_cagr()* and *p_cagr()* are functions of calculating the expected returns of the option trade using the stock return (r), option contract premium (c00 and p00), moneyness (x), and the contract expiration period (d). The expected return is annualized by multiplying a factor 252/d. *CS()* and *PS()* are functions of computing the call and put option premiums using the Black-Scholes option price formula. *dec()* is a function to restrict a real number to three decimal points.

```
retn=function(x){len=length(x);retn=vector(length=len); for (i in 2:len)
retn[i]=x[i]/x[i-1]-1; return(retn)}
prz=function(r){len=length(r);prz=vector(length=len); prz[1]=1;for (i in 2:len)
prz[i]=prz[i-1]*(1+r[i-1]); return(prz)}
c_cagr=function(r,c00,x,d){g=r; for (i in 2:length(r)) {if (r[i]<x) {g[i]=c00+r[i]} else
{g[i]=c00+min(0,x)} }; return(mean(g)*252/d)}
p_cagr=function(r,p00,x,d){g=r; for (i in 2:length(r)) {if (r[i]<x) {g[i]=p00+r[i]-
max(0,x)} else {g[i]=p00} }; return(mean(g)*252/d)}
CS=function(t,r,sigma,x){w1=(r*t-log(1+x))/(sigma*sqrt(t)); w2=sigma*sqrt(t)/2
return((pnorm(w1+w2)-(1+x)*pnorm(w1-w2)/exp(r*t)))}
PS=function(t,r,sigma,x){w1=(r*t-log(1+x))/(sigma*sqrt(t)); w2=sigma*sqrt(t)/2
return((1+x)*pnorm(w2-w1)/exp(r*t)-pnorm(-w1-w2))}
dec=function(y){return(sprintf("%.3f",y))}

#tkr="SPY Model"; mu=0.0004472206; std=0.01210494
tkr="QQQ Model"; mu=0.0005332306; std=0.01710912
n=500000; sigma=std*sqrt(252); c0=prz(rnorm(n,mu,std)); r=0.04; d=c(1,5,20)
x=seq(from=-0.2,to=0.2, by=0.001); cagr=x; c00=x; p00=x

CC=function(d){
  c1=c0[seq_along(c0) %% d == 0]; r1=retn(c1)
  for (i in 1:length(x)){
    c00[i]=CS(d/252,r,sigma,x[i]); cagr[i]=c_cagr(r1,c00[i],x[i],d)}
  return(cagr)
}

CSP=function(d){
  c1=c0[seq_along(c0) %% d == 0]; r1=retn(c1)
  for (i in 1:length(x)){
    p00[i]=PS(d/252,r,sigma,x[i]); cagr[i]=p_cagr(r1,p00[i],x[i],d)}
  return(cagr)
}

st=proc.time()
cagr=lapply(d,CC)
```

```
#cagr=lapply(d,CSP)
proc.time()-st

plot(x,cagr[[1]],type="l", main=paste(tkr,": Max Gain = ",
    dec(max(unlist(cagr)))), xlab=expression(x[K]),ylab="Returns"); grid(lty=2)
lines(x,cagr[[2]], col="blue"); lines(x,cagr[[3]], col="red")
```

Constant *tkr* is the model name; *mu* and *std* are the mean and standard deviation of the daily returns. 500,000 simulated daily returns are generated using the *rnorm()* function and subsequently converted to prices using the *prz()* function. Three option expiration days (1, 5 and 20) are in the vector *d*.

Variable *x* is the moneyness taking values from -0.2 to +0.2, cagr the vector for the expected returns of the option trade, and c00 (or p00) the vector for the option premiums.

CC and CSP are functions that organize the price vectors in accordance with the option contract expiration period (d) so that only the prices at the beginning and end of the period are used to compute the stock price return, *r1*. For each moneyness, *x*, the option contract premium is computed and followed by the expected annualized return, *cagr*.

The statement “`cagr=lapply(d,CC)`” computes the expected return of option trade for each d value. Finally, the results are presented in a plot. The statement “`proc.time()-st`” is to show the computation time elapses with the very large sample size.

<u>CSP Trading Algorithm</u>

For ITM contracts, with no stock ownership change, the gain at expiration is simply the premium ( $PS(x_K)$). With ownership assignment, however, one retains the premium but receives the stock now priced at $S(< S_0)$ which is less than $S_0$. Therefore there is a loss of $x_S(< 0)$ that is partially compensated by the strike price. For OTM contracts, with no ownership assignment, the gain at expiration is the same as that of the ITM contracts without assignment. However, if an assignment occurs, one keeps the premium, but the stock price change ($S - S_0$) is a negative number; therefore one received the assigned stock at a loss.

| Moneyness | Ownership Assignment | Trading Returns |
|---|---|---|
| ITM ($x_K > 0$) | No assignment | $PS(x_K)$ |
| | With assignment | $PS(x_K) + x_S - x_K$ |
| OTM ($x_K < 0$) | No assignment | $PS(x_K)$ |
| | With assignment | $PS(x_K) + x_S$ |

Combining both scenarios of the moneyness situations, the algorithm becomes:

- No ownership assignment: $gain = PS(x_K)$
- With ownership assignment: $gain = PS(x_K) + x_S - max(x_K, 0)$

<u>Simulation Results</u>

Figure 1 shows the simulated expected (or mean) annual returns from the covered call option trading of SPY. The black curve is for daily contract expiration, blue curve for weekly expiration, and red curve for monthly expiration.

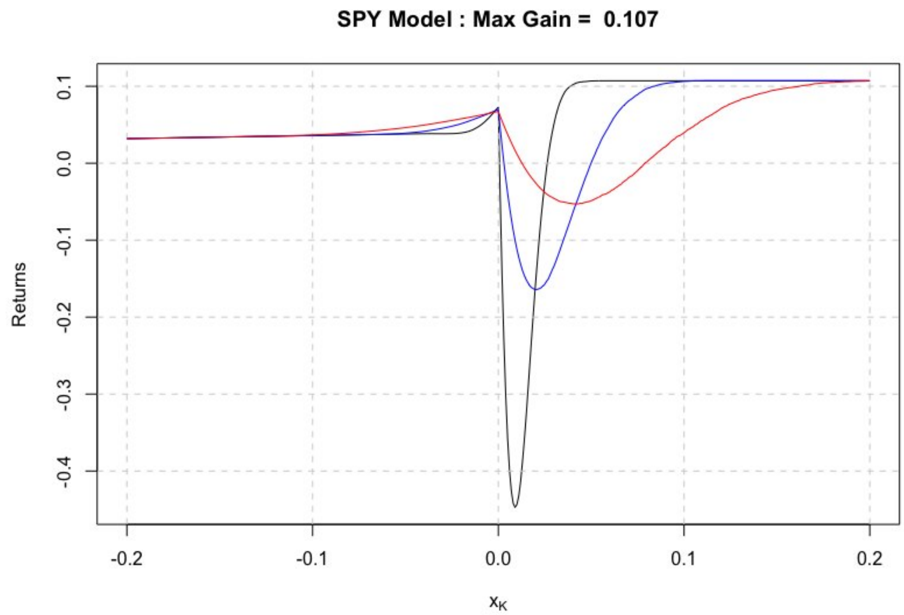


*Figure 1. Simulated Covered Call Option Trade of SPY for Daily (black), Weekly (blue), and Monthly (red) Expirations*

Figure 2 shows the simulated expected annual returns of the covered call option trading of QQQ. The black curve is for daily contract expiration, blue curve for weekly expiration, and red curve for monthly expiration.

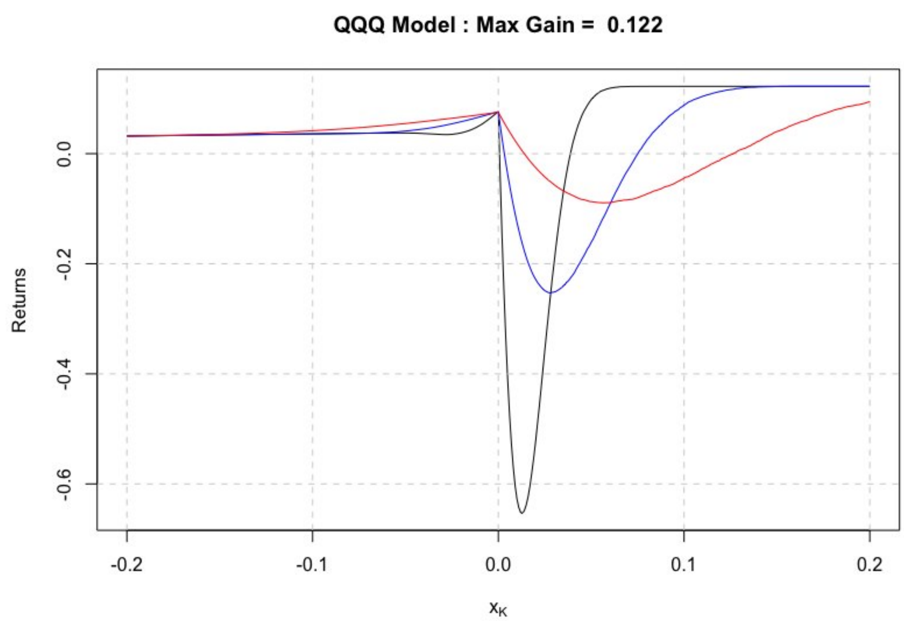


*Figure 2. Simulated Covered Call Option Trade of QQQ for Daily (black), Weekly (blue), and Monthly (red) Expirations*

These two plots are consistent with the simulations with the historical data in general structures (see “Simulation of Covered Call Option Trade”) except that several fin structures in the data simulations are not seen in the Monte Carlo simulations. The reasons for missing these fine structures are not known at this time.

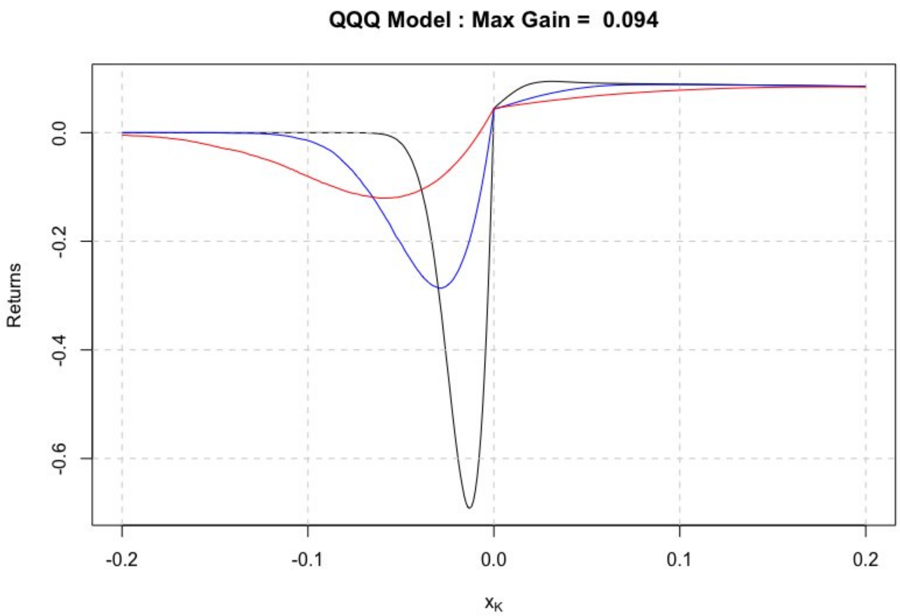


*Figure 3. Simulated Cash-Secured Put (CSP) Option Trade of QQQ for Daily (black), Weekly (blue), and Monthly (red) Expirations*

Figure 3 is the Monte Carlo simulation of the QQQ CSP trades. On the left side of the plot no positive profit is found where the strike price is far lower than the stock price ($x_K < 0$) as such a stock assignment never occurs. The lack of profit is due to the fact that the capital has been “secured away” from earning an interest on one hand, and the received “premium” from the trade becomes exceedingly tiny on the other. On the far right side the plot, the stock assignment always happens. One collects the sizable premium for the trade but the stock received is always at a loss from the original price.

## Concluding Remarks

Covered calls and cash secured put options trades are not mathematically profitable “strategies” that can be deployed for making consistent excess returns. Rather, they could be consistently unprofitable. By extension, the so-called “Wheel Strategy” is not a mathematically profitable strategy either. Instead, both covered call and cash-secured put option trades may serve as a tactical tool during a specific phase of the market cycle. For example, at a market peak, a cover call option might be awarded with a profitable premium without the stocks being assigned away since the contract expiration will be during a market downturn. Conversely, a cash-secured put option could earn a sound premium at a market bottom with little risk of being forced upon the stocks at a loss for the assignment since the stock price at the contract expiration would be lifted above the bottom.