Project #4: Put-call parity. More Monte Carlo. The normal approximation to the binomial.

Samuel Kalisch

2024-04-16

```r
library(ggplot2)
library(tidyverse)
library(gridExtra)
library(grid)
```

Problem #1 (25 points)

Put-call parity

(5 points) Using put-call parity, find the formula for the interest rate if all other ingredients are known.

$$V_c - V_p = S_0 - e^{-rT}K$$

$$e^{-rT}K = S_0 - [V_c - V_p]$$

$$e^{-rT} = \frac{S_0 - [V_c - V_p]}{K}$$

$$-rT = \ln\left[\frac{S_0 - [V_c - V_p]}{K}\right]$$

$$r = \frac{1}{T} \cdot \ln\left[\frac{K}{S_0 - [V_c - V_p]}\right]$$

(5 points) Based on the above, what can you say about the interest rates r you expect to obtain for varying values of the strike price?

Based on the put-call parity formula, the interest rate (r) depends on the relationship between the current price of the underlying asset (S_0), the prices of the call (V_c) and put (V_p) options, and the strike price (K). This happens because V_c and V_p are functions of the strike. Clearly, as K increases, the numerator will increase. The interesting thing will happen with the denominator. Let's denote $V_c - V_p$ as the call premium. Our formula above yields:

$$r = \frac{1}{T} \cdot \ln \left[\frac{K}{S_0 - \text{Call Premium}} \right]$$

Analysis Based on the Value of Strike Price K

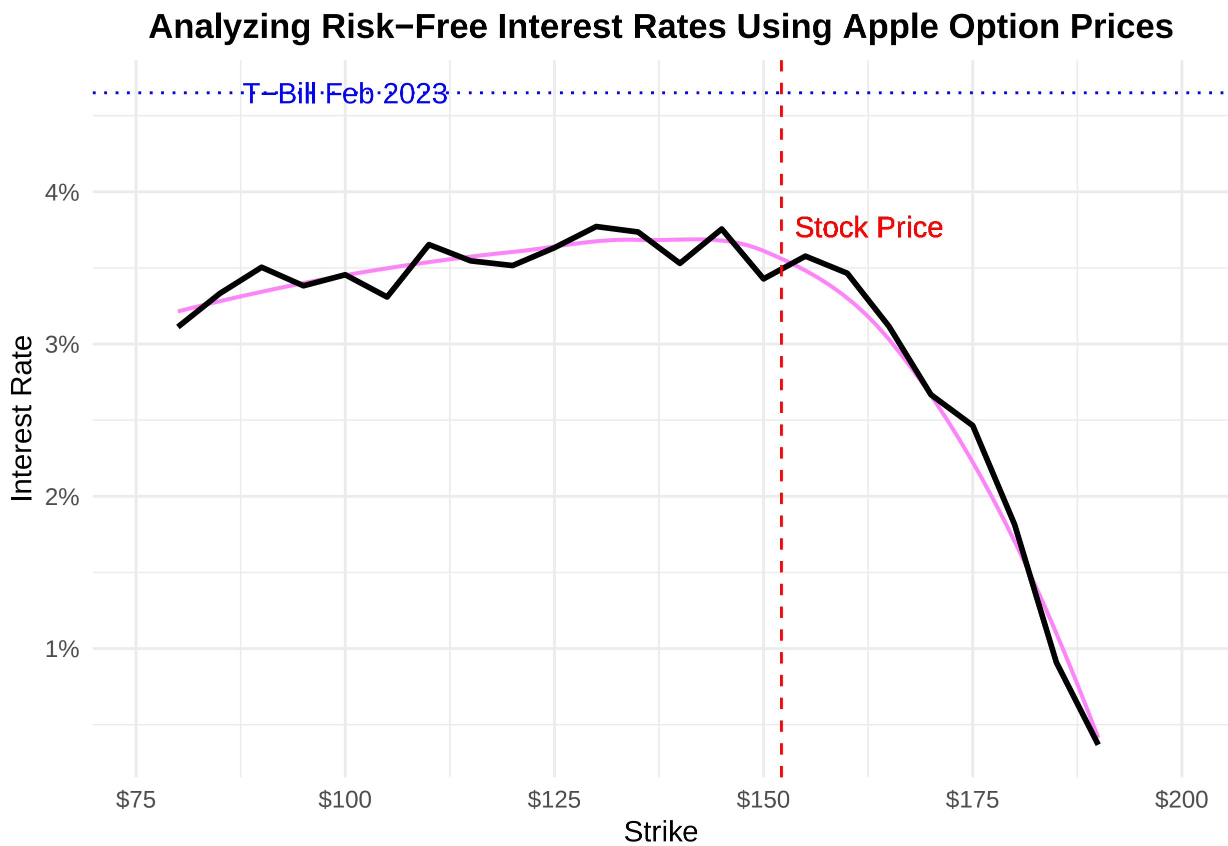
1. **When the strike price (K) is lower than the price (S_0) (In-the-Money Options):**
   - Call options are more expensive relative to put options, making the call premium positive and potentially large (due to the lognormal distribution under BSM).
   - The denominator $S_0 - [V_c - V_p]$ becomes smaller, making the fraction $\frac{K}{S_0 - \text{Call Premium}}$ smaller. This could lead to a negative value inside the logarithm, translating to lower or even negative values for r.
2. **When the strike price (K) is close to the current price (S_0) (At-the-Money Options):**
   - Call and put options are priced more similarly, leading to a smaller call premium.
   - The term $\frac{K}{S_0 - \text{Call Premium}}$ approaches 1 as K approaches S_0, leading to r approaching zero. This reflects a balanced market view.
3. **When the strike price (K) is higher than S_0 (Out-of-the-Money Options):**
   - Call options are cheaper relative to puts, making the call premium smaller or negative.
   - While the fraction $\frac{K}{S_0 - \text{Call Premium}}$ can be greater than 1 initially (suggesting a positive r), as K significantly exceeds S_0, the impact of a decreasing V_c and an increasing V_p can moderate or reverse the trend, potentially leading to a more moderate or even lower r. This adjustment reflects the lessened impact of the call premium and the increased pricing of risk associated with puts in these scenarios.

(10 points) Based on the data set "apple-parity.csv", calculate the continuously compounded, risk-free interest rate for all the strike prices given. Be careful about how you use the *bid* and *ask/offer* prices of the options. Set $T = 0.25$. Plot the values of the interest rate you obtain as they depend on the strike prices.

FRED (3 Month T-bill)

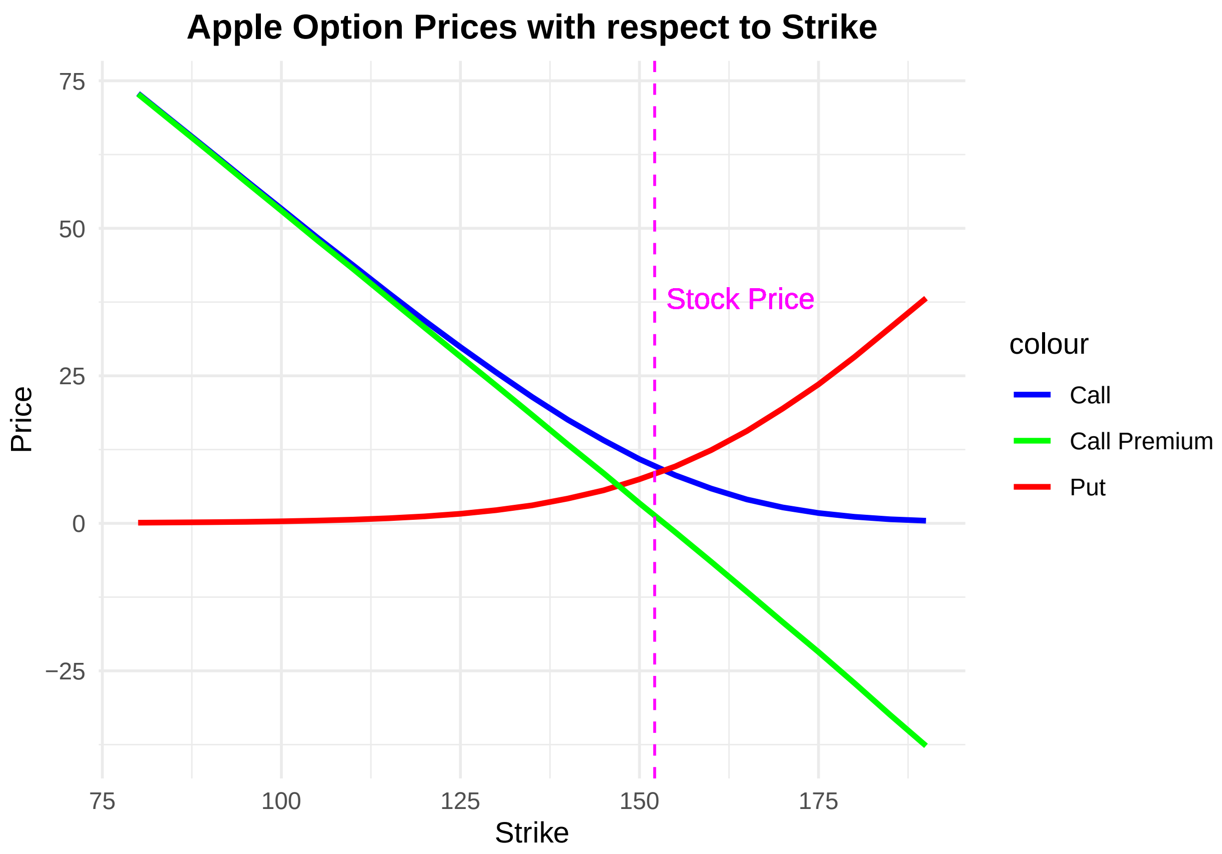
```
#find r
calculate_r <- function(Vc0, Vp0, S0, K, T= .25) {return(log(K/(Vp0 - Vc0 + S0)) / T)}

#read dataset
appl_parity = read.csv("apple-parity.csv")

#wrangle dataset
appl_parity= appl_parity%>%
  mutate(Put = (BidPut+AskPut)/2)%>%
  mutate(Call = (BidCall+AskCall)/2)%>%
  mutate(r = calculate_r(Call,Put,Stock,Strike))

# Fit a smoothed curve
smoothed_curve <- geom_smooth(data = appl_parity, aes(x = Strike, y = r*100),
  method = "loess",formula = y ~ x, se = FALSE, color = "orchid1", linewidth = .7)

#plotting
ggplot(appl_parity, aes(x = Strike)) + smoothed_curve+
geom_line(aes(y = r*100), linewidth = 1) +
labs(title = "Analyzing Risk-Free Interest Rates Using Apple Option Prices",
x = "Strike",
y = "Interest Rate") +
```

```r
theme_minimal()+theme(plot.title = element_text(hjust = 0.5, face = "bold"))+
  scale_x_continuous(labels = scales::dollar_format(big.mark = ","),
                     limits = c(.95*min(appl_parity$Strike),1.05*max(appl_parity$Strike)))+
  scale_y_continuous(labels = scales::percent_format(scale = 1, accuracy = 1))+
  geom_vline(xintercept = appl_parity$Stock, color = "red", linetype = "dashed") +
  annotate("text", x = appl_parity$Stock + 10.5, y = 100*max(appl_parity$r), label = "Stock Price",
           color = "red") +
   geom_hline(yintercept = 4.65, linetype = "dotted",
          color = "blue") +  # Adding the horizontal line for T-Bill 3 moth
  annotate("text", x =100, y = 4.65, label = "T-Bill Feb 2023", color = "blue")
```




```r
#extra plot
ggplot(appl_parity, aes(x = Strike)) +
  geom_line(aes(y = Call, color = "Call"), linewidth = 1) +
  geom_line(aes(y = Put, color = "Put"), linewidth = 1) +
  geom_line(aes(y = (Call-Put), color = "Call Premium"), linewidth = 1) +
  geom_vline(xintercept = appl_parity$Stock, color = "magenta", linetype = "dashed") +
  annotate("text", x = appl_parity$Stock + 12, y = max(appl_parity$Put), label = "Stock Price",
           color = "magenta") +
  labs(
    title = paste("Apple Option Prices with respect to Strike"),
    x = "Strike",
    y = "Price"
  ) +
  theme_minimal() +
```

```r
theme(plot.title = element_text(hjust = 0.5, face = "bold")) +
scale_color_manual(
  values = c("blue", "green", "red"),
  labels = c("Call", "Call Premium","Put")
)
```



(5 points) What do you notice about the above plot? Does it or does it not agree with your prediction from the second question in the problem? Substantiate your answer.

Observations of the Plot

The plot provided depicts implied interest rates derived from Apple's option prices, with a reference to the T-Bill rate from February 2023. From the graph, we can observe:

- **Comparison with T-Bill Rate**: The implied rates for strike prices are one point lower than with T-Bill rate, which stands at 4.75%. However, the rates for higher strike prices deviate significantly below the T-Bill rate, indicating a divergence in market expectations for these options
- **Decline Beyond Stock Price**: As the strike prices exceed the current stock price (indicated by the red dashed vertical line), there is a noticeable decline in the implied interest rates. The rate of decline is gradual at first but then becomes more pronounced.
- **Sharp Decrease for Higher Strikes**: For strike prices significantly higher than the stock price, the implied interest rates drop sharply. This steep decline, particularly after the strike price of $150.

Comparison with Predictions

The plot of implied interest rates from Apple's option prices alongside the 3-month T-bill rate presents some interesting points:

The provided plot of implied interest rates from Apple's option prices relative to the 3-month T-bill rate showcases several patterns:

1. **In-the-Money Options**: My prediction indicated that for in-the-money options, the implied interest rates might be lower due to a large positive call premium. The graph does show this decrease.The implied rates stay near or below the T-bill rate.
2. **At-the-Money Options**:As expected, the implied interest rates around the at-the-money strikes are at it's highest, however, they are significantly lower than the T-Bill rate.
3. **Out-of-the-Money Options (Strike price K is higher than S_0)**: Tbe prediction was for a potentially more moderate or even lower r as the strike price significantly exceeds S_0. The plot shows that the implied interest rates decrease significantly for higher strikes, which agrees with the prediction.

Problem #2 (25 points)

To what do you attribute the discrepancies you observed in the previous problem? Think about it a bit. Then, if you don't have any ideas look at the paper by *Brenner* and *Galai* uploaded into Canvas (under *Files* in the folder *articles*).

Insights from Brenner and Galai on Implied Interest Rates

Brenner and Galai's study on implied interest rates brings to light several key factors that can cause discrepancies between market-derived implied interest rates and those predicted by theoretical models. Their research highlights the complexities introduced by American options, dividends, transaction costs, market inefficiencies, and imperfect market conditions.

- **American vs. European Options:** American options, which can be exercised before expiration, differ from European options that can only be exercised at maturity. This can affect the values of puts and calls (if unprotected from dividends), hence the implied interest rates.
- **Dividends:** Dividends, which are not accounted for in the basic put-call parity equation, can lead to early exercise, especially for in-the-money call options. This can lead to deviations from the expected implied interest rates.
- **Transaction Costs and Short Sale Restrictions:** The put-call parity assumes no transaction costs and full use of short sale proceeds. Real-world restrictions bias the implied interest rates since in practice only a portion of the short sale proceeds can be used.
- **Market Inefficiencies:** Discrepancies between borrowing and lending rates are expected to be small in a well-functioning market, but inefficiencies can maintain this spread, causing deviations in the implied rates.
- **Imperfect Market Conditions:** Non-synchronous trading, price quote discreteness, bid-ask spreads, and other market frictions can lead to deviations from theoretical models.

Options markets provide competitive rates that generally align with market trends. However, factors such as early exercise and unequal treatment of short and long positions can cause significant discrepancies. The implied rates tend to be closer to the borrowing rate than the lending rate, suggesting a bias in the cost of capital derived from options markets.

Problem #3 (15 points)

Let the continuously compounded, risk-free interest rate be 0.05.

Consider a stock whose current price is $80 and whose volatility is 0.2. We will be pricing a variety of options using a *forward binomial tree*.

```
r = 0.05
s0 = 80
sigma = .2
```

(5 points) Price a one-year, $85-strike European call option analytically using a $100-$period binomial tree.

```
T=1
n = 100
h =T/n
K = 85

#factors
u=exp(r*h+sigma*sqrt(h))
d=exp(r*h-sigma*sqrt(h))

#risk-neutral probability
p=(exp(r*h)-d)/(u-d)

#possible payoffs
k=n:0
s.T = s0*u^k*d^(n-k)
v.T=pmax(s.T-K,0)

#the risk-neutral probabilties of reaching the terminal nodes
rn.probs = dbinom(k, size=n, prob=p)

#risk-netral price
v.0=exp(-r*T)*sum(v.T*rn.probs)
v.0
## [1] 5.997075
```

(5 points) Price a one-year, $85-strike European call option using *Monte Carlo* with 10000 simulations with a $100-$period binomial tree.

```
T=1
n = 100
h =T/n
K = 85
nsims=10^4

#factors
u=exp(r*h+sigma*sqrt(h))
d=exp(r*h-sigma*sqrt(h))

#risk-neutral probability
p=(exp(r*h)-d)/(u-d)
```

```
#montecarlo simulation for number of upsteps
x=rbinom(nsims, size=n,prob=p)

#simulated payoff
s.T = s0*u^x*d^(n-x)
v.T=pmax(s.T-K,0)

#average of simulated payoffs
v.bar=mean(v.T)

#Monte-Carlo price
v.mc = exp(-r*T)*v.bar
v.mc
## [1] 6.084751
```

(5 points) Price a half-year, \$78-strike European put option analytically using a 100−period binomial tree.

```
T=.5
n = 100
h =T/n
K = 78

#factors
u=exp(r*h+sigma*sqrt(h))
d=exp(r*h-sigma*sqrt(h))

#risk-neutral probability
p=(exp(r*h)-d)/(u-d)

#possible payoffs
k=n:0
s.T = s0*u^k*d^(n-k)
v.T=pmax(K-s.T,0)

#the risk-neutral probabilties of reaching the terminal nodes
rn.probs = dbinom(k, size=n, prob=p)

#risk-netral price
v.0=exp(-r*T)*sum(v.T*rn.probs)
v.0
## [1] 2.714835
```

(5 points) Price a half-year, \$78-strike European put option using *Monte Carlo* with 10000 simulations with a 100−period binomial tree.

```
T=.5
n = 100
h =T/n
K = 78
nsims=10^4

#factors
u=exp(r*h+sigma*sqrt(h))
```

```r
d=exp(r*h-sigma*sqrt(h))

#risk-neutral probability
p=(exp(r*h)-d)/(u-d)


#montecarlo simulation for number of upsteps
x=rbinom(nsims, size=n,prob=p)

#simulated payoff
s.T = s0*u^x*d^(n-x)
v.T=pmax(K-s.T,0)

#average of simulated payoffs
v.bar=mean(v.T)

#Monte-Carlo price
v.mc = exp(-r*T)*v.bar
v.mc
## [1] 2.66892
```

(5 points) Comment on the accuracy of the *Monte Carlo* method. Which theorem from probability is useful here?

Law of Large Numbers (LLN)

The Law of Large Numbers (LLN) is fundamental in understanding the accuracy of the Monte Carlo method. It states that as the number of trials or samples (n) increases, the sample average converges to the expected value. Mathematically, for a sequence of independent and identically distributed random variables $X_1, X_2, \ldots, X_n$ with mean μ and finite variance, the LLN is expressed as:

- $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i = \mu$

By the properties of variance, $Var(\bar{X}_n) = \frac{\sigma^2}{n}$. As n approaches infinity, $Var(\bar{X}_n)$ approaches zero. This implies that $\bar{X}_n$ becomes increasingly concentrated around its mean μ as n increases, leading to the convergence of $\bar{X}_n$ to μ as n tends to infinity, as stated by the LLN.

Problem #4 (25 points)

Let $\{X_n, n = 1, 2, \ldots\}$ be a sequence of random variables such that

$$X_n \sim Binomial(n, p)$$

where p is a constant between 0 and 1.

(5 points) State the *DeMoivre-Laplace Theorem* (aka the *normal approximation to the binomial*) in the context of the above sequence of random variables.

DeMoivre-Laplace Theorem

Since X_n denotes the number of successes that occur when n independent trials, each resulting in a success with probability p, are performed, then,

- $E[X_n] = np$

- $Var[X_n] = np(1-p) \Rightarrow SD[X_n] = \sqrt{np(1-p)}$

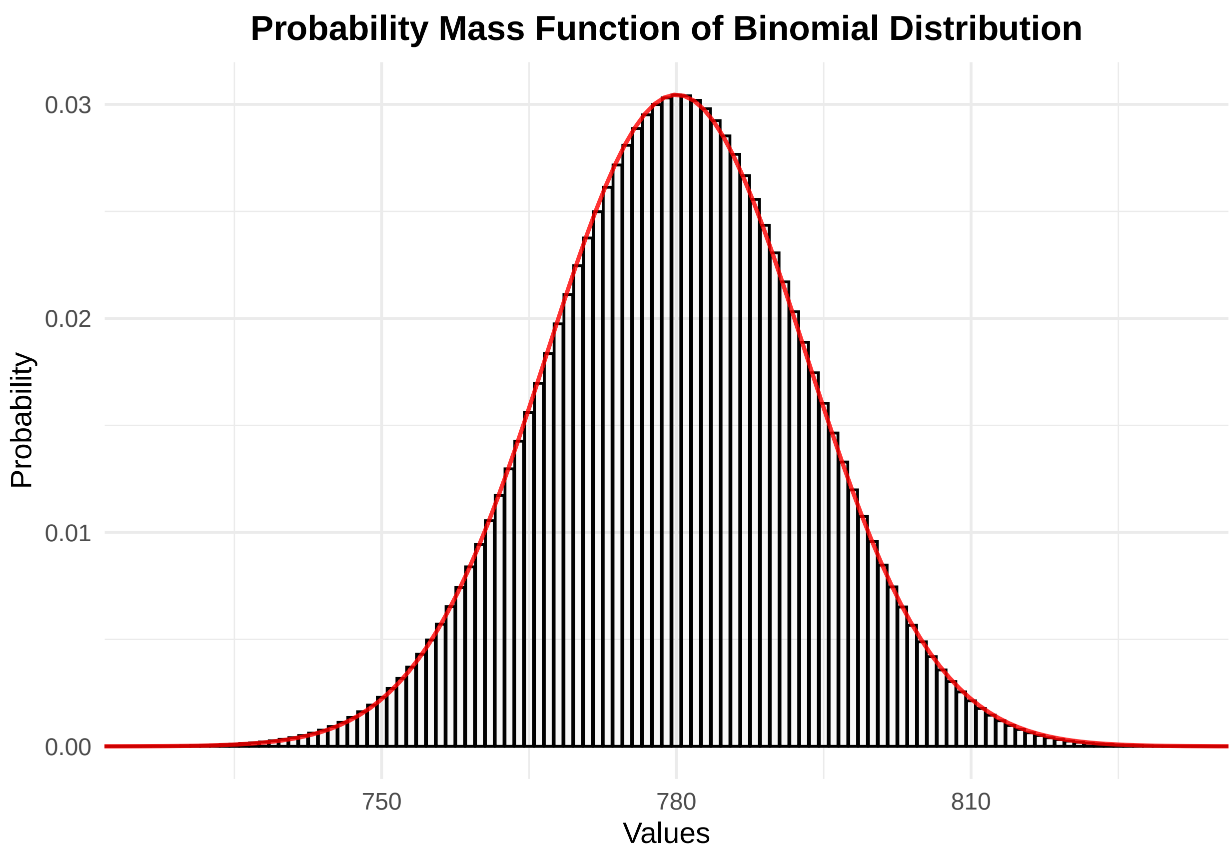
$$\frac{X_n - np}{\sqrt{np(1-p)}} \xrightarrow[n\to\infty]{\mathcal{D}} \mathcal{N}(0,1)$$

Also, for any $a < b$,

$$P(a < X_n \leq b) \approx \mathcal{N}\left(\frac{b-np}{\sqrt{np(1-p)}}\right) - \mathcal{N}\left(\frac{a-np}{\sqrt{np(1-p)}}\right)$$

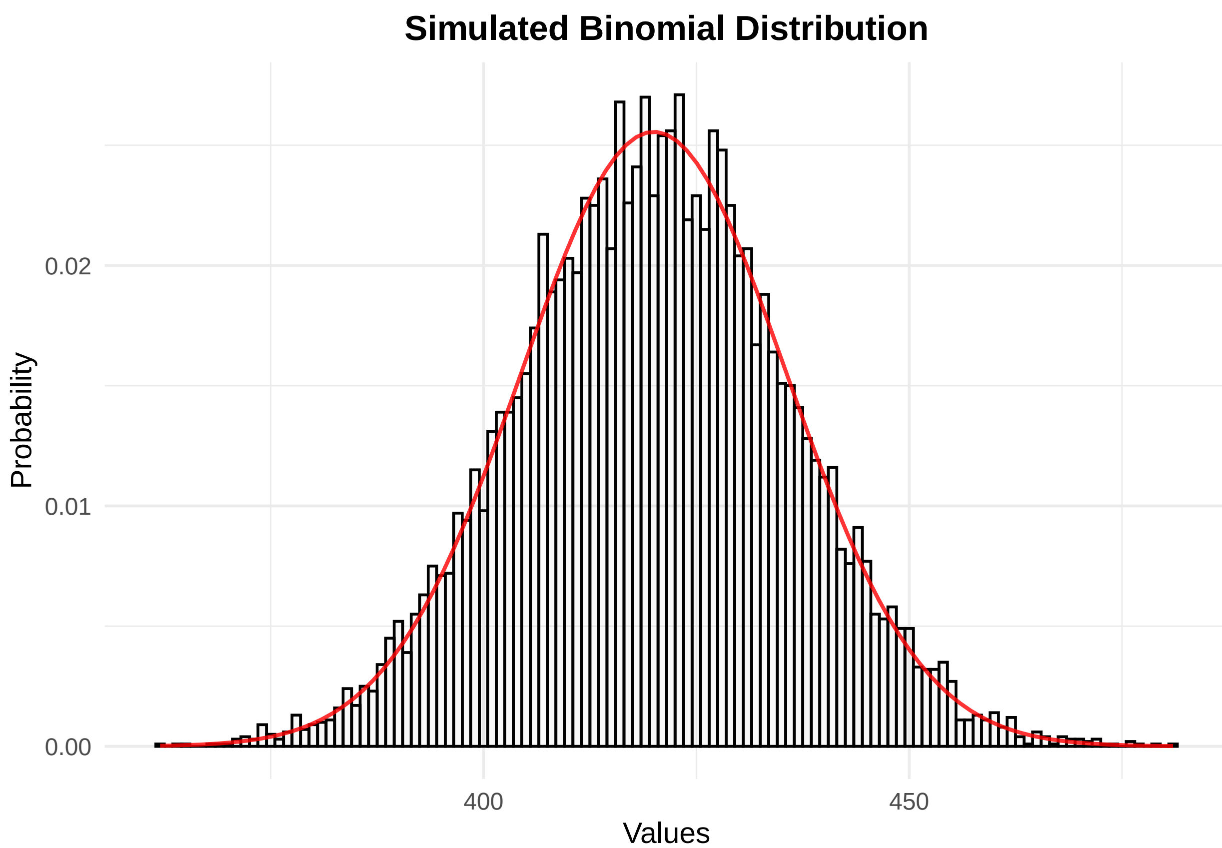
(5 points) Let $p = 0.78$. For $n = 1000$, plot the **theoretical** histogram of X_n. Superimpose the appropriate density of the normal distribution on that histogram (according to the theorem referenced above).

```
p = 0.78
n = 1000
k = n:0
probabilities <- dbinom(k, size = n, prob = p)

#specify parameters for DML theorem
mu = n*p
sd = sqrt(n*p*(1-p))

#find relevant range of distribution
min_val <- min(k[probabilities > 0.00001])
max_val <- max(k[probabilities > 0.00001])

#calculate density for normal dist
x_values <- seq(0, n, length.out = 1000)
normal_density <- dnorm(x_values, mean = mu, sd = sd)

#plotting
ggplot(data.frame(x = k, probability = probabilities), aes(x = x, y = probability)) +
  geom_bar(stat = "identity", fill = "grey", color = "black", alpha = 0.1)+
  coord_cartesian(xlim = c(min_val, max_val)) +
  geom_line(data = data.frame(x = x_values, density = normal_density), aes(x = x, y = density),
            color = "red", linewidth = .75, alpha = 0.8) +
  labs(title = "Probability Mass Function of Binomial Distribution",
       x = "Values",
       y = "Probability") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



(5 points) Let $p = 0.42$. For $n = 1000$, draw 10000 simulated values of X_n and plot the histogram of the draws. Superimpose the appropriate density of the normal distribution on that histogram (according to the theorem referenced above).

```
p = 0.42
n = 1000
n.sims = 10^4

#monte carlo for binomial
binomial <- rbinom(n.sims, size = n, prob = p)

#specify parameters for DML theorem
mu = n*p
std = sqrt(n*p*(1-p))

#plotting
ggplot(data.frame(x = binomial), aes(x = x)) +
  geom_histogram(binwidth = 1, aes(y = after_stat(density)),
                 fill = "grey", color = "black", alpha = 0.1)+
  stat_function(fun = dnorm, args = list(mean = mu, sd = std),
                color = "red", linewidth = .7, alpha = .8) +
  labs(title = "Simulated Binomial Distribution",
       x = "Values",
       y = "Probability") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```



Problem #5 (10 points)

Let $\{Y_n, n = 1, 2, \dots\}$ be a sequence of random variables such that

$$Y_n \sim Binomial(n, p_n)$$

where p_n is given by

$$p_n = \frac{1}{1 + e^{0.25\sqrt{1/n}}}$$

.

For $n = 100, 1000, 5000, 10000$, draw 10000 simulated values of Y_n and plot the histogram of the draws. Does the theorem referenced in the previous problem apply to this situation or not? Substantiate your answer.

```
#probability function
prob<- function(n){return((1+exp(.25*sqrt(1/n)))^-1)}

#parameters for normal distribution
munormal <- function(n){return(n*prob(n))}
sigmanormal <- function(n){return(sqrt(n*prob(n)*(1-prob(n))))}

#monte-carlo simulation function
generate_rbinomial <- function(n.sims, ns, ps,i){return(rbinom(n.sims,
                                                   size = ns[i], prob = ps[i]))}
#plotting function
plotsim<- function(binomial, mu, sigma, n){
```

```
  ggplot(data.frame(x = binomial), aes(x = x)) +
  geom_histogram(binwidth = 1, aes(y = after_stat(density)),
                 fill = "grey", color = "black", alpha = 0.1)+
  stat_function(fun = dnorm, args = list(mean = mu, sd = sigma), color = "red",
                linewidth = .7, alpha = .8) +
  labs(title = paste("n =", format(n, big.mark = ",")),
       x = "Values",
       y = "Probability") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, face = "italic", size = 10))
}
```

```
ns = c(100,1000,5000,10000)
ps = prob(ns)
mu = munormal(ns)
sigma = sigmanormal(ns)
n.sims = 10^6

#monte-carlo for binomials
binomialmatrix = list(rep(NA, r))
for(i in 1:4){binomialmatrix[[i]] = generate_rbinomial(n.sims, ns, ps,i)}
```

Does it apply?

```
#check if applies
check_theorem<-function(n, p){
  bool = (n*p>=10 & n*(1-p)>=10)
  if(bool){return("valid:")}
  if(!bool){return("invalid:")}
 }

for(i in 1:4){
  cat("The thorem for",format(ns[i], big.mark = ","),"simulated values is",
      check_theorem(ns[i], ps[i]),"\n")
  cat("n*p =",ns[i]*ps[i],"\n")
  cat("n*(1-p) =",ns[i]*ps[i],"\n","\n")
}
## The thorem for 100 simulated values is valid:
## n*p = 49.37503
## n*(1-p) = 49.37503
##
## The thorem for 1,000 simulated values is valid:
## n*p = 498.0236
## n*(1-p) = 498.0236
##
## The thorem for 5,000 simulated values is valid:
## n*p = 2495.581
## n*(1-p) = 2495.581
##
## The thorem for 10,000 simulated values is valid:
## n*p = 4993.75
## n*(1-p) = 4993.75
##
```

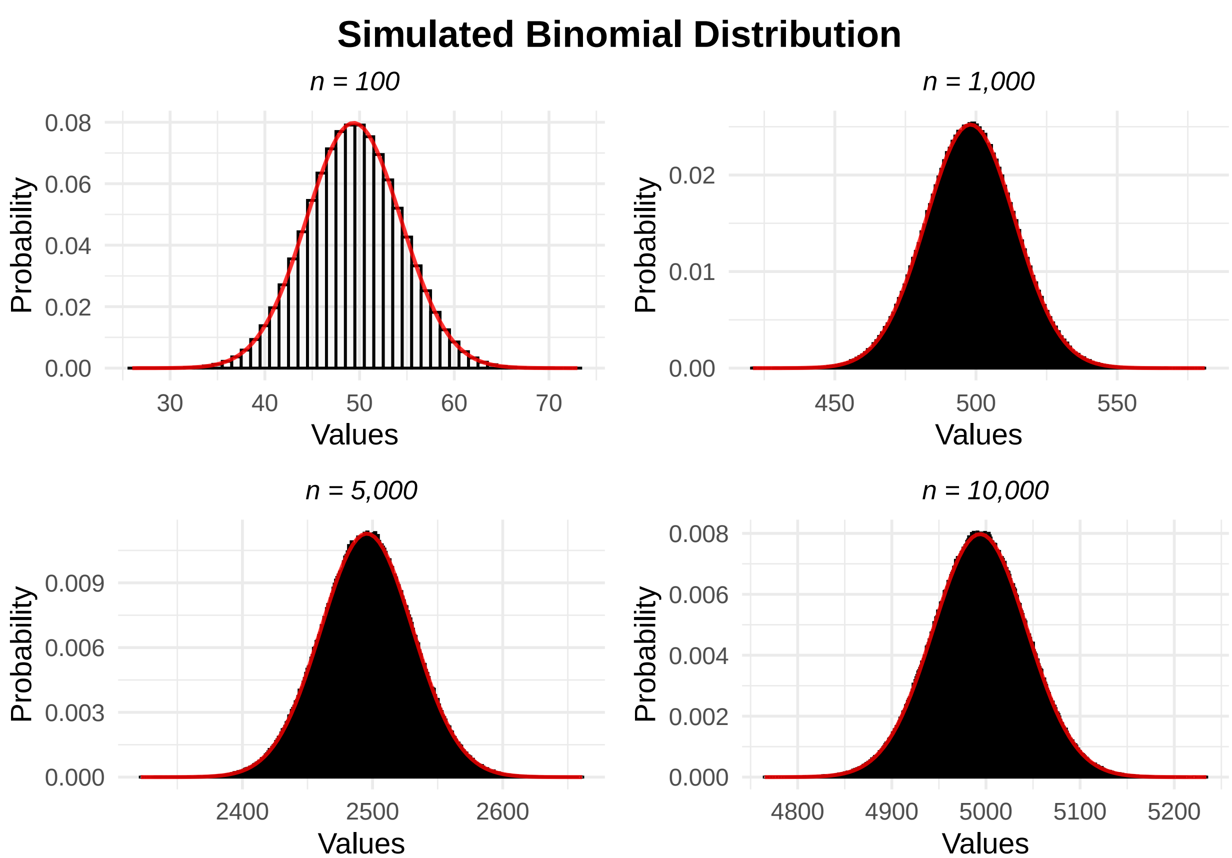
Conclusions:

- The DeMoivre-Laplace Theorem is valid for all tested values of n $(100, 1000, 5000, and 10000)$.
- Each calculation confirms that np and $n(1-p)$ are comfortably greater than 10, meeting the requirement for the theorem's applicability.
- The number of simulations (10000) doesn't affect the validity of the theorem but impacts the precision of the histogram representation of the distribution.

Based on the analysis, we can conclude that the DeMoivre-Laplace Theorem applies to the sequence of random variables Y_n described in the problem statement. The conditions for the theorem's applicability are met for all tested values of n, and increasing the number of simulations improves the accuracy of the histogram representation without altering the validity of the theorem.

Plottting:

```
grid.arrange(
  plotsim(binomialmatrix[[1]], mu[1],sigma[1], ns[1]),
  plotsim(binomialmatrix[[2]], mu[2],sigma[2], ns[2]),
  plotsim(binomialmatrix[[3]], mu[3],sigma[3], ns[3]),
  plotsim(binomialmatrix[[4]], mu[4],sigma[4],ns[4]),
  nrow = 2, ncol = 2,
  top = textGrob("Simulated Binomial Distribution",
                 gp = gpar(fontsize = 14, fontface = "bold"))
  )
```



“‘