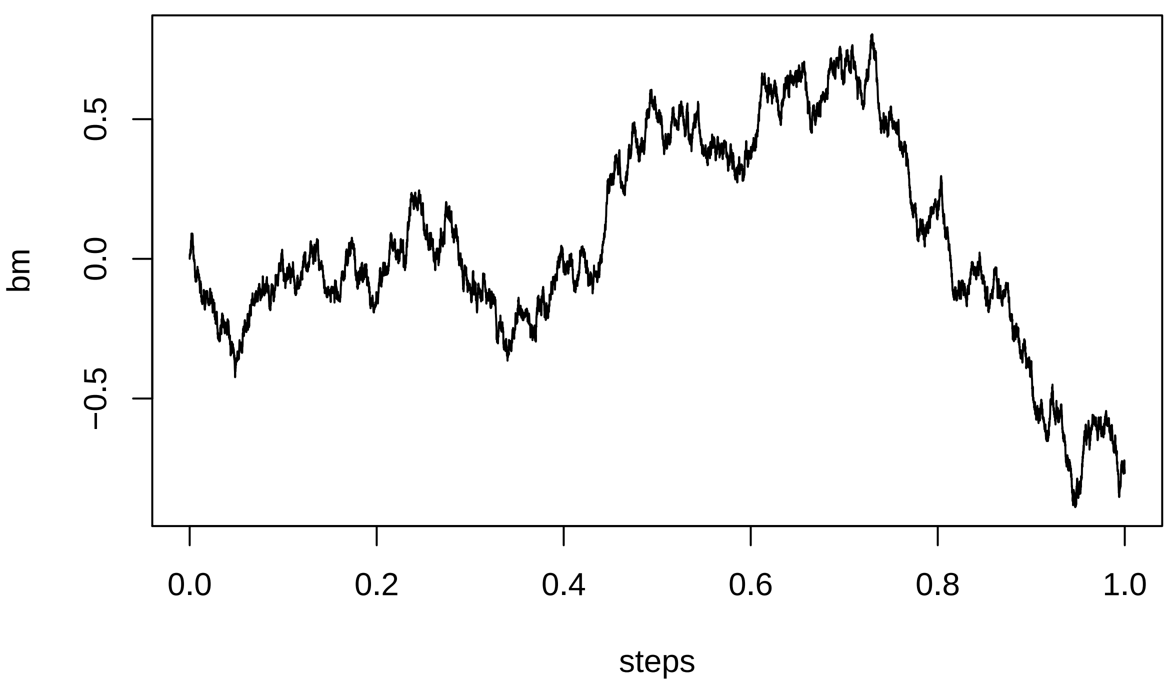
Brownian Motion & Option Pricing

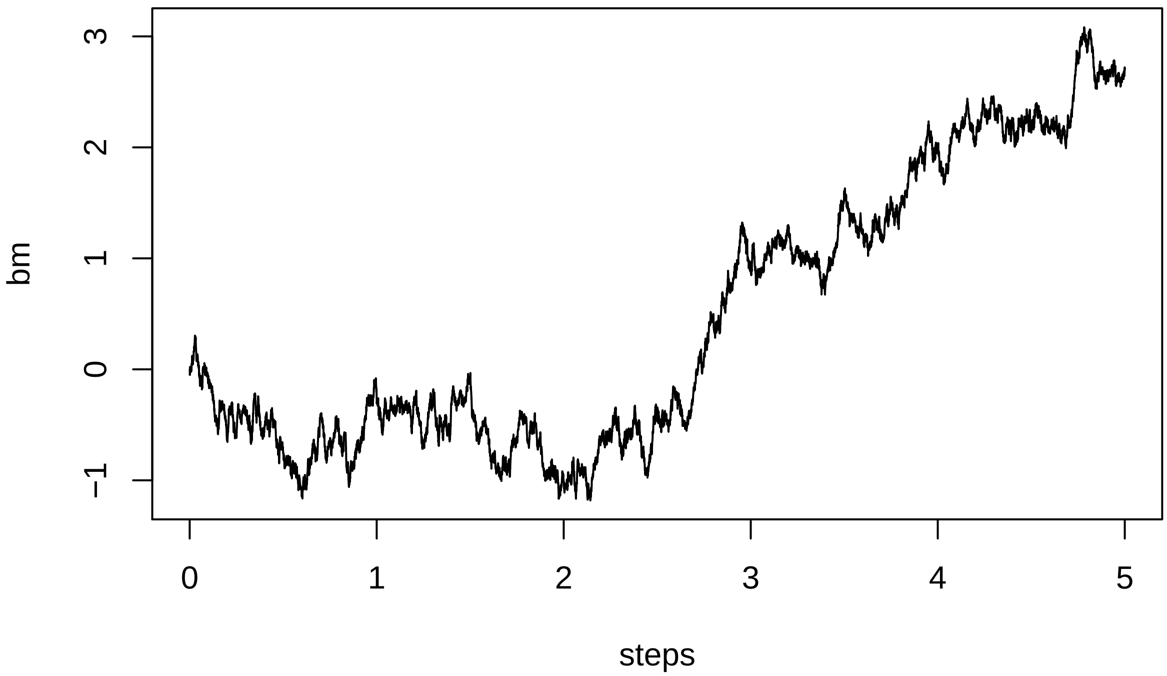
Aymen Rumi

Functions to Simulate Brownian Motion & Brownian Motion with Drift

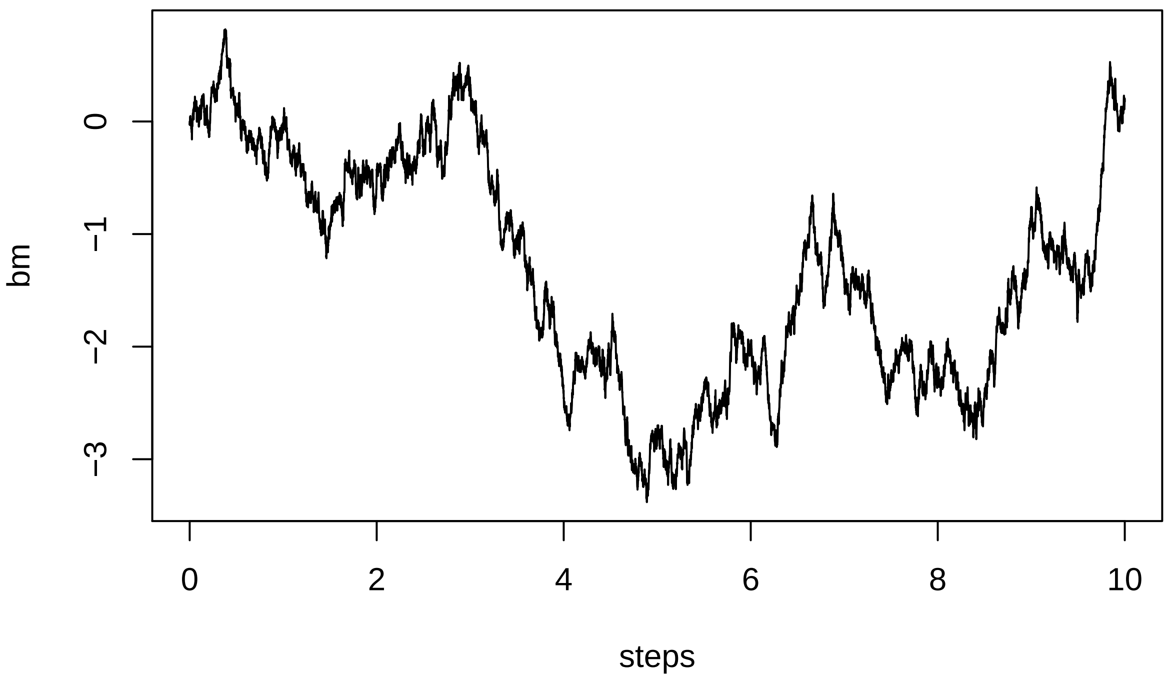
```r
# Brownian Motion
SimulateBrownianMotion<-function(t,n)
{
    bm<-c(0,cumsum(rnorm(n,0,sqrt(t/n))))
    steps<-seq(0,t,length=n+1)
    plot(steps,bm,type="l")
}

# Simulations 1: t=1
SimulateBrownianMotion(1,10000)
```




```r
# Simulations 2: t=5
SimulateBrownianMotion(5,10000)
```

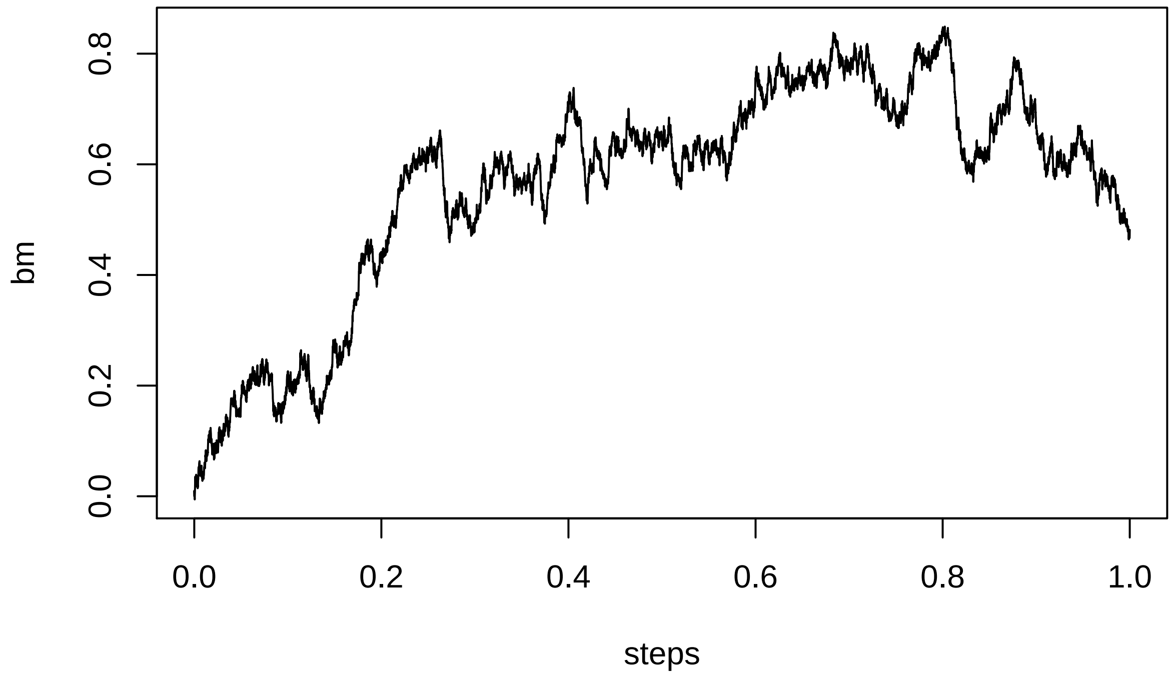



```r
# Simulations 3: t=10
SimulateBrownianMotion(10,10000)
```

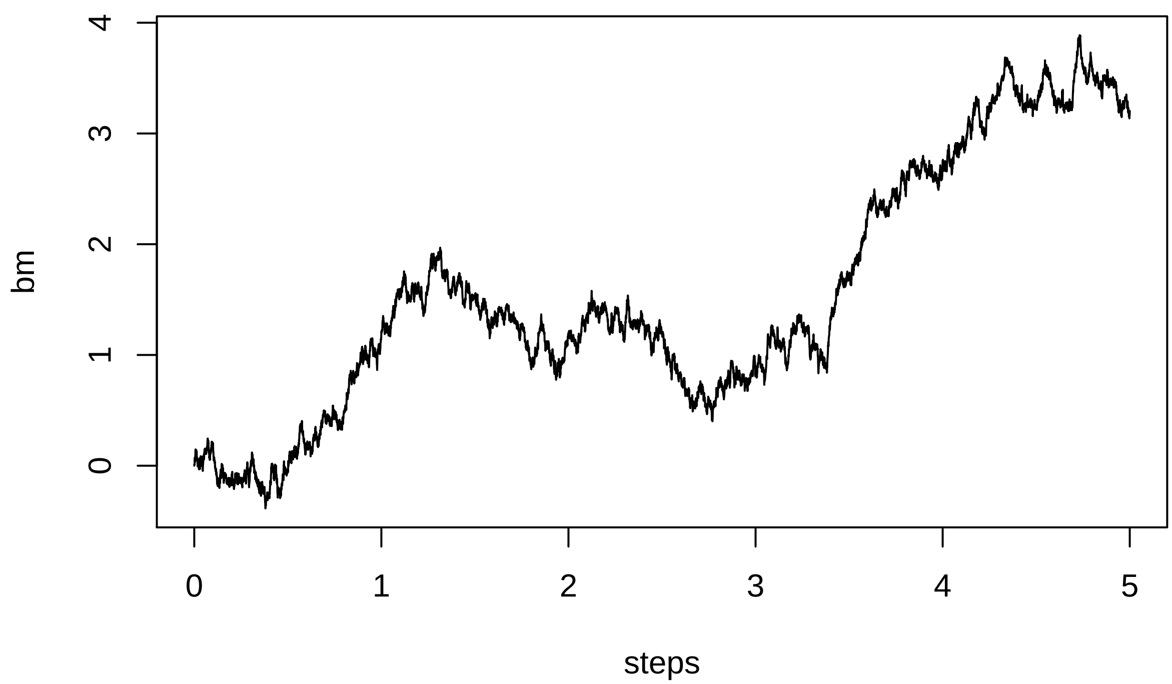



```r
# Brownian Motion with Drift
SimulateBrownianMotionWithDrift<-function(t,n,mu,sigma)
{
    bm<-c(0,cumsum(rnorm(n,(mu*(t/n)),sqrt(sigma*(t/n)))))
    steps<-seq(0,t,length=n+1)
    plot(steps,bm,type="l")
}
```

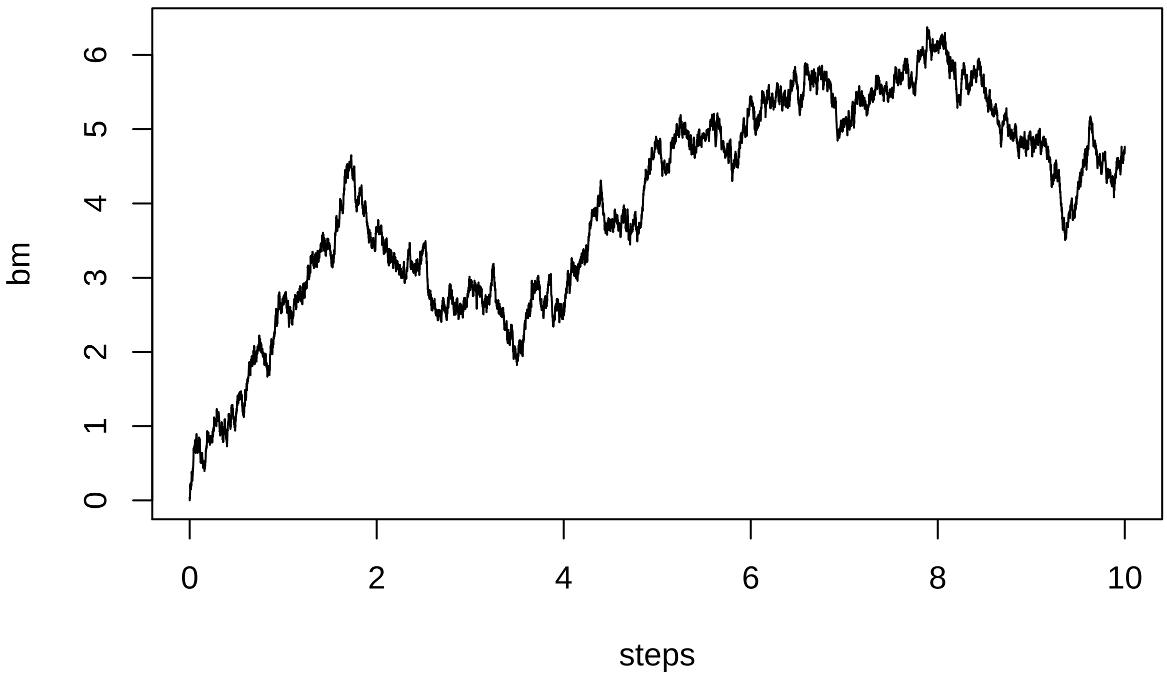
```r
# Simulations 1: t=1, mean=0.1, sigma=0.25
SimulateBrownianMotionWithDrift(1,10000,0.1,0.25)
```




```r
# Simulations 2: t=5, mean=0.25, sigma=0.75
SimulateBrownianMotionWithDrift(5,10000,0.25,0.75)
```




```r
# Simulations 2: t=10, mean=0.5, sigma=1.25
SimulateBrownianMotionWithDrift(10,10000,0.5,1.25)
```

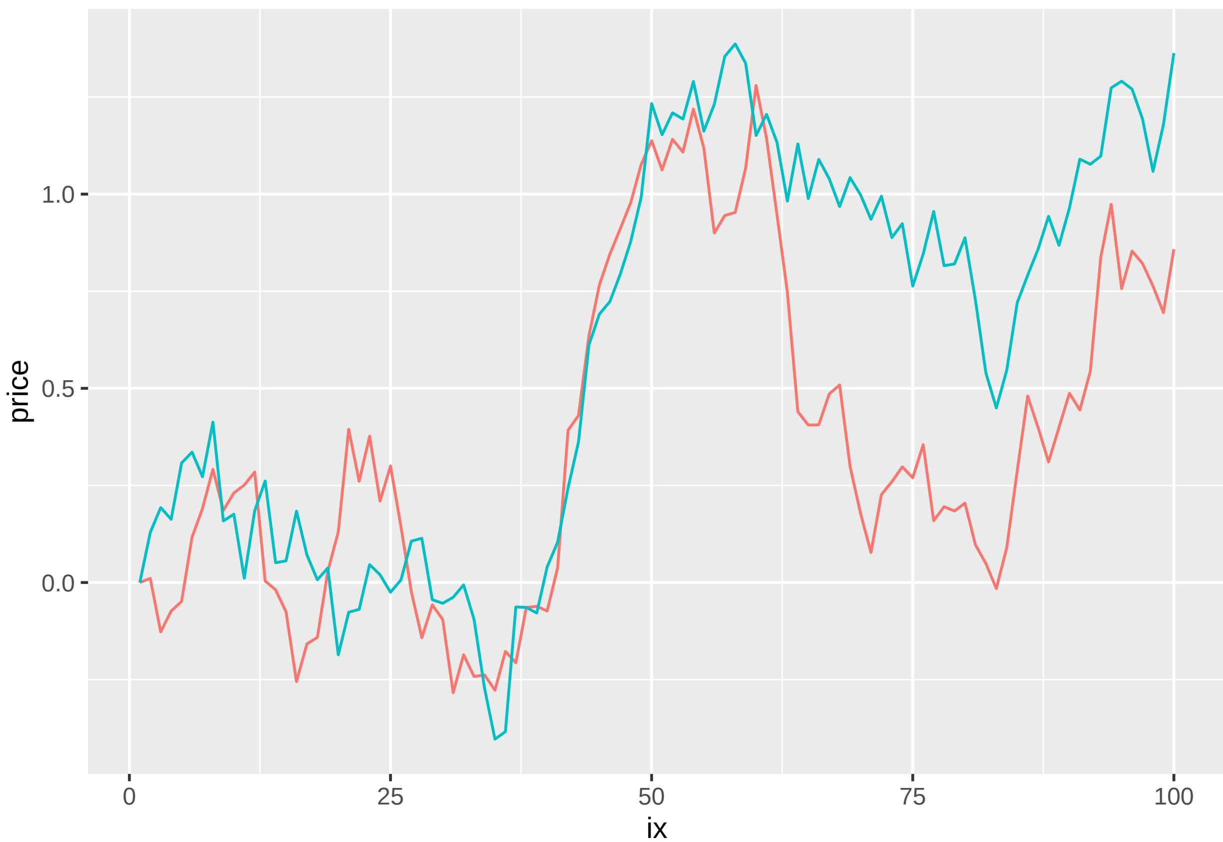


Repeated Brownian Motion Simulations: Geometric & BM with Drift

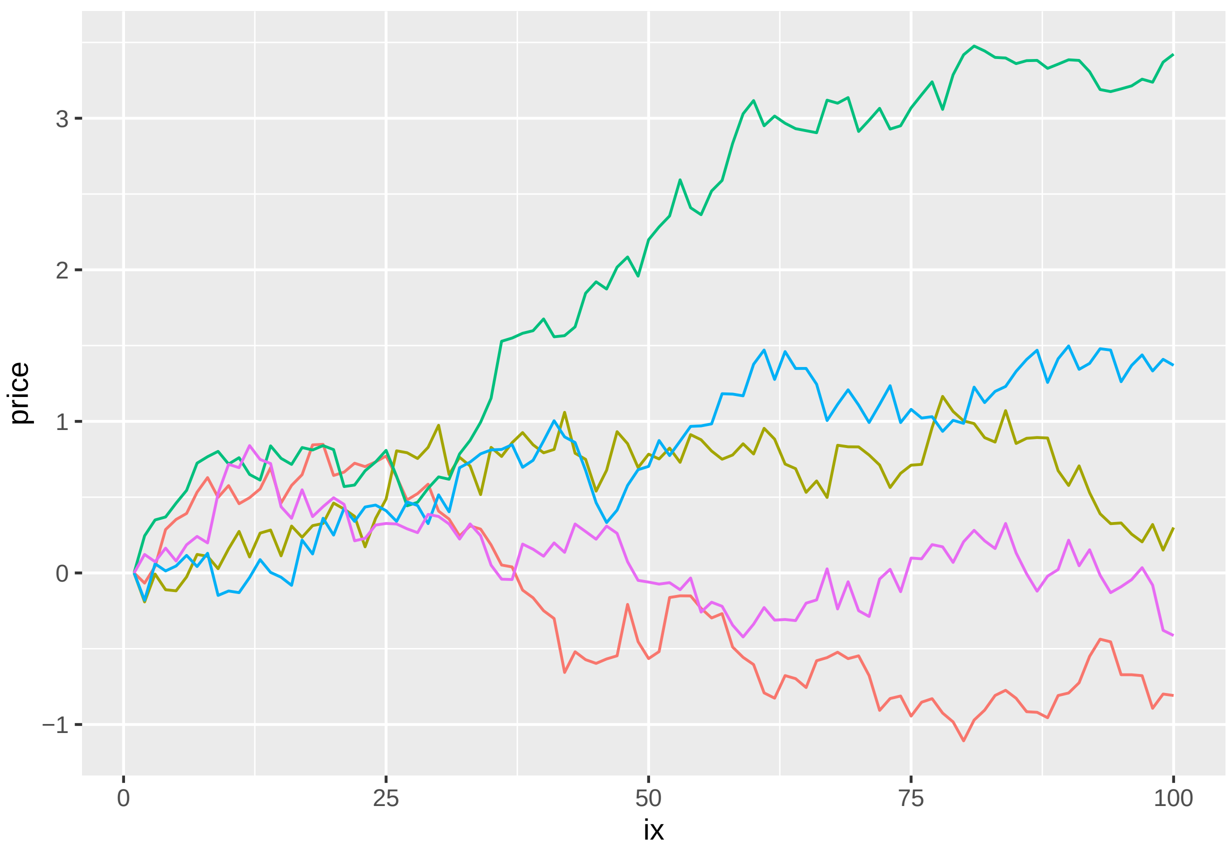
You can also embed plots, for example:

```
# Brownian Motion with Drift
BrownianMotionWithDrift_Simulation<-function(nsim,t,n,mu,sigma)
{

    gbm <- matrix(ncol = nsim, nrow = n)

    for (simu in 1:nsim) {

        gbm[1, simu] <- 0

        for (day in 2:n) {


            gbm[day, simu] <- gbm[(day-1), simu] + rnorm(1,(mu*(t/n)),sqrt(sigma*(t/n)))

        }
    }


    gbm_df <- as.data.frame(gbm) %>%
    mutate(ix = 1:nrow(gbm)) %>%
    pivot_longer(-ix, names_to = 'sim', values_to = 'price')


    ggplot(data=gbm_df,aes(x=ix, y=price, color=sim)) +
```

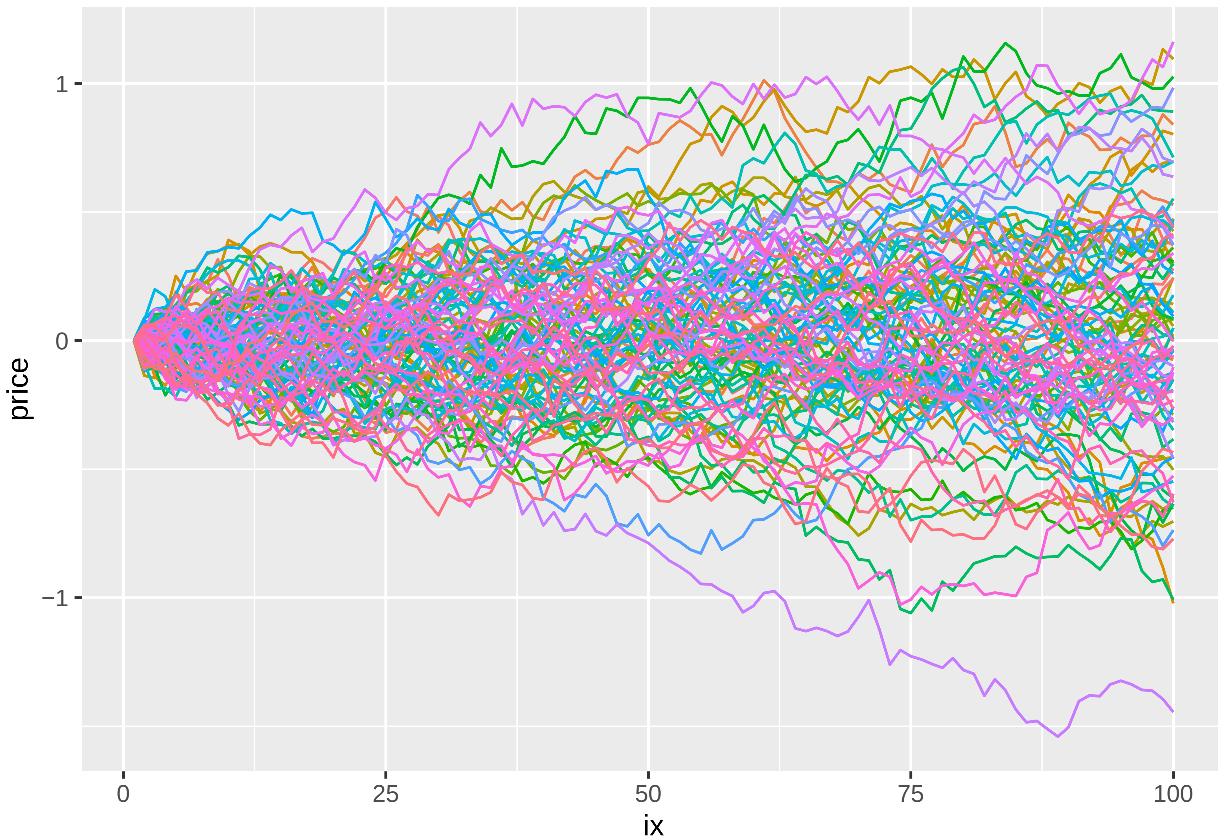
```
    geom_line() +
    theme(legend.position = 'none')

}

# Simulation 1
BrownianMotionWithDrift_Simulation(2,0.5,100,2,3)
```




```
# Simulation 2
BrownianMotionWithDrift_Simulation(5,0.5,100,2,3)
```

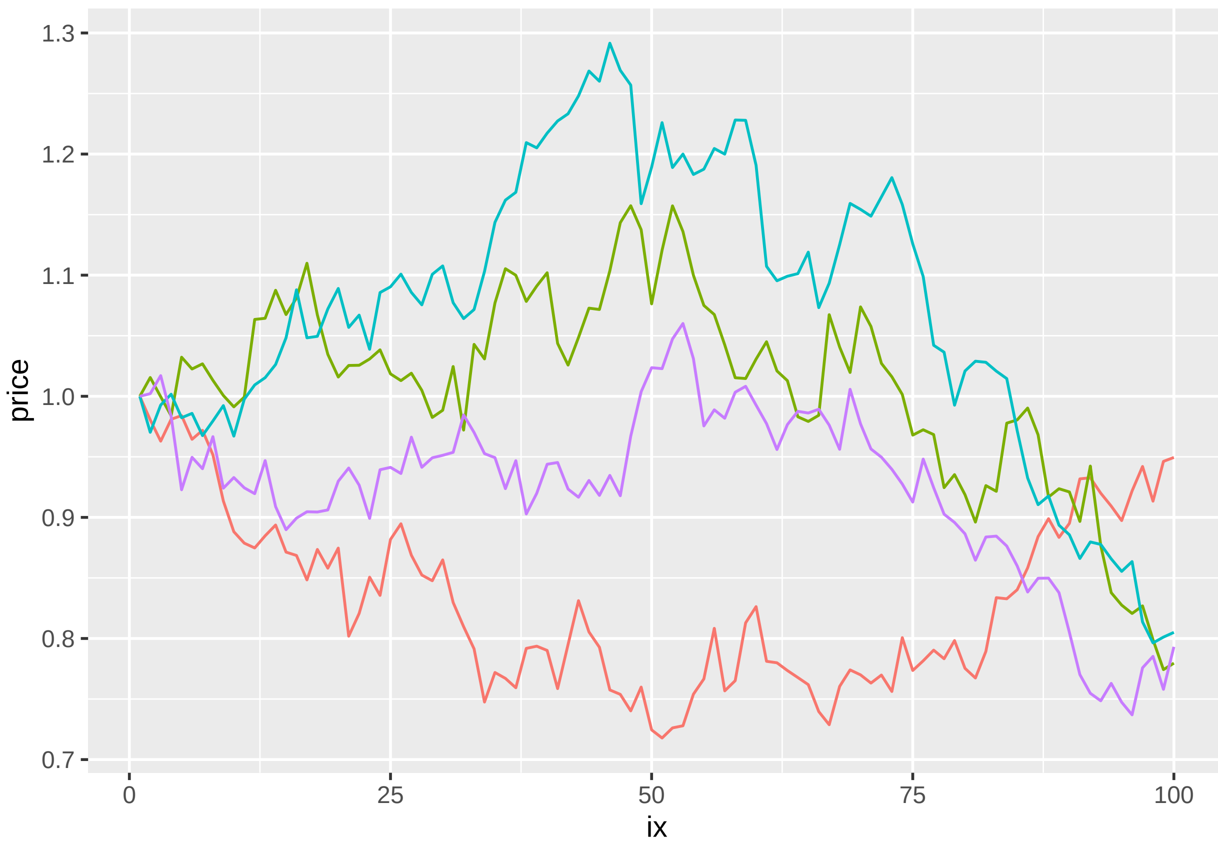



```
# Simulation 3
BrownianMotionWithDrift_Simulation(100,0.5,100,0,0.5)
```

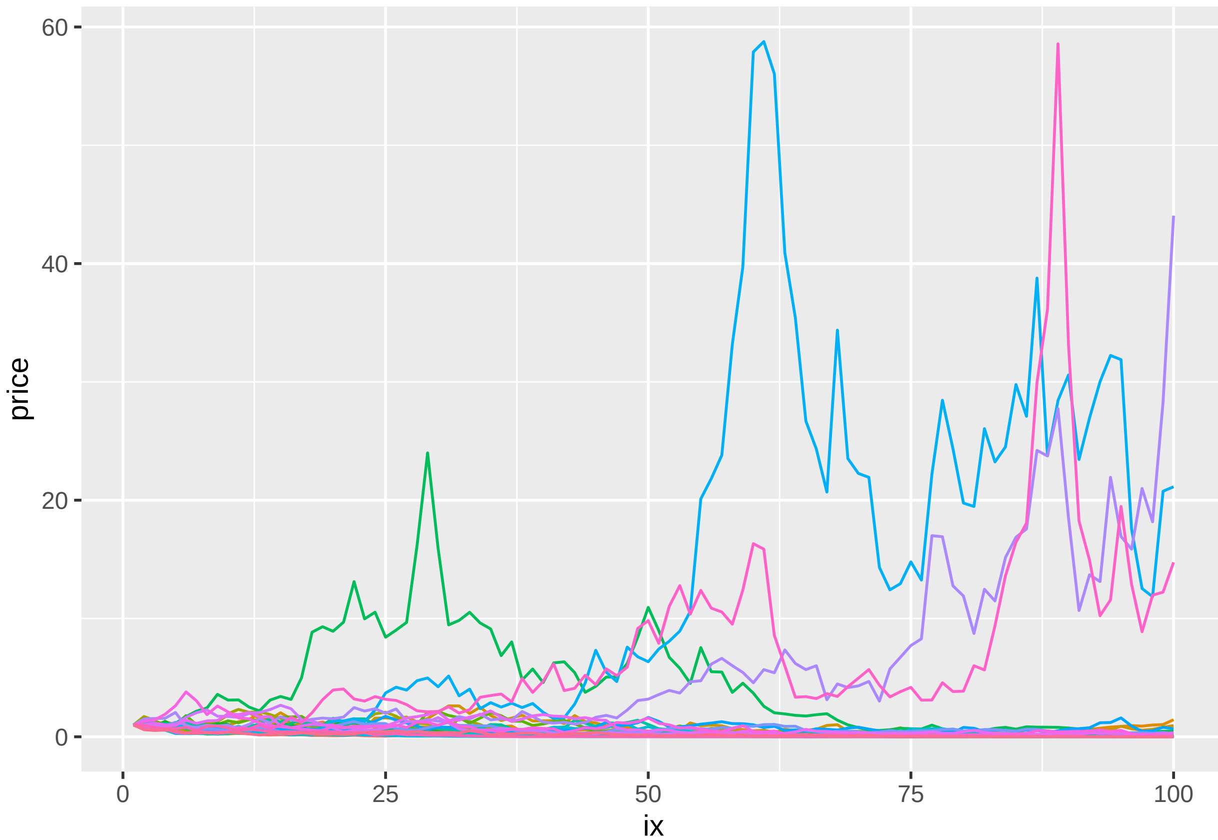



```r
# Geometric Brownian Motion: Simulation of Stock Prices
GeometricBrownianMotion_Simulation<-function(nsim,t,mu,sigma,S0)
{

    gbm <- matrix(ncol = nsim, nrow = t)

    for (simu in 1:nsim) {

    gbm[1, simu] <- S0
    for (day in 2:t) {

      epsilon <- rnorm(1)
      dt = 1 / 365
      gbm[day, simu] <- gbm[(day-1), simu] * exp((mu - sigma * sigma / 2) * dt + sigma * epsilon * sqrt

        }
    }

    gbm_df <- as.data.frame(gbm) %>%
    mutate(ix = 1:nrow(gbm)) %>%
    pivot_longer(-ix, names_to = 'sim', values_to = 'price')


    ggplot(data=gbm_df,aes(x=ix, y=price, color=sim)) +
    geom_line() +
    theme(legend.position = 'none')
```

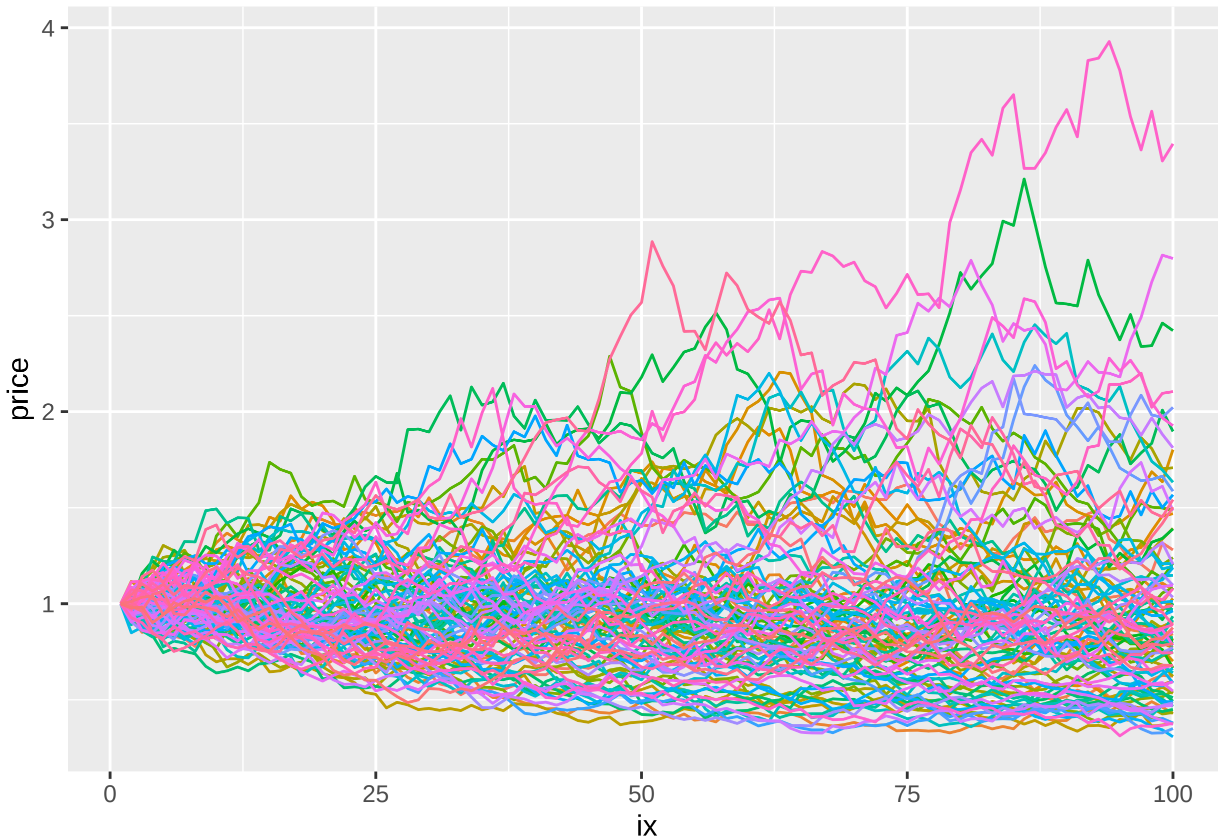
```
}

# Sample 1: mean=0, sigma=0.5
GeometricBrownianMotion_Simulation(4,100,0,0.5,1)
```




```
# Sample 2: mean=1, sigma=5
GeometricBrownianMotion_Simulation(25,100,1,5,1)
```



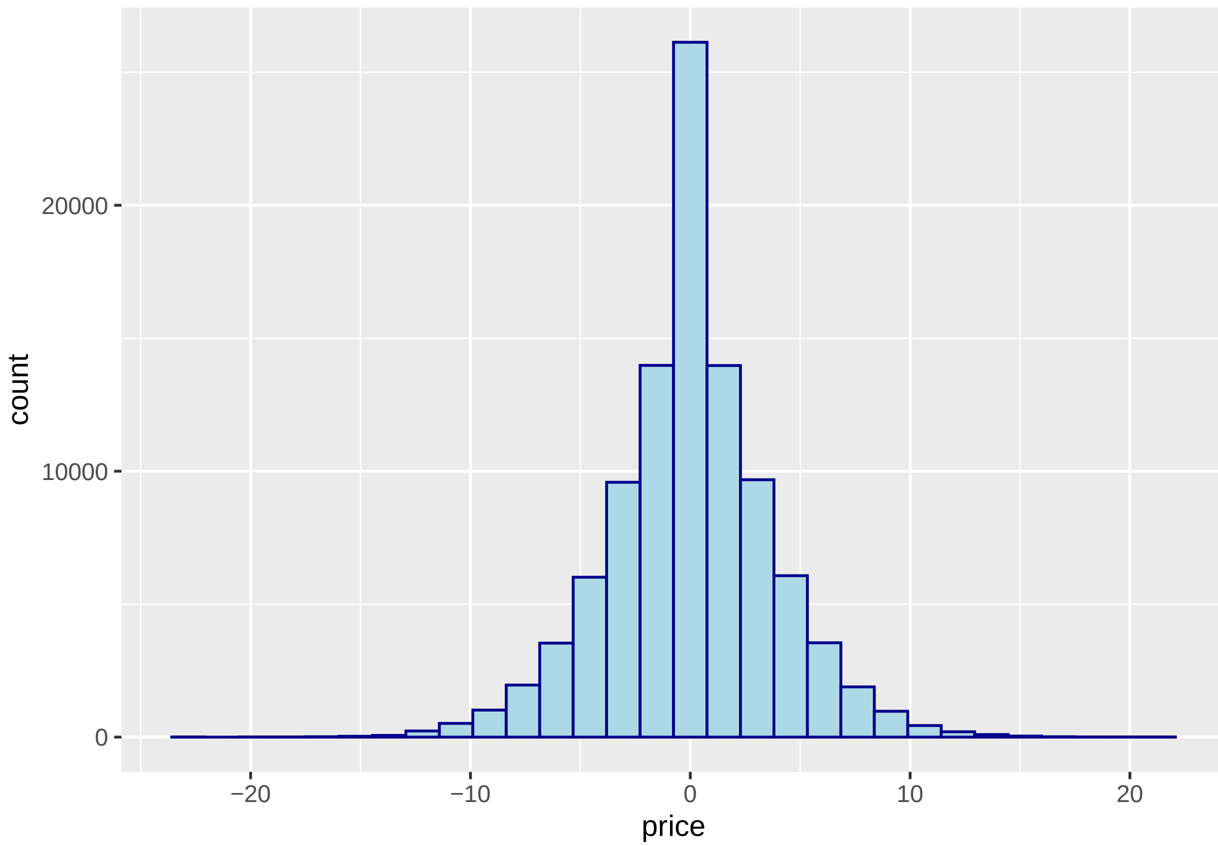

```r
# Sample 3: mean=0, sigma=1
GeometricBrownianMotion_Simulation(100,100,0,1,1)
```



Plotting Brownian Motion Simulation Histograms

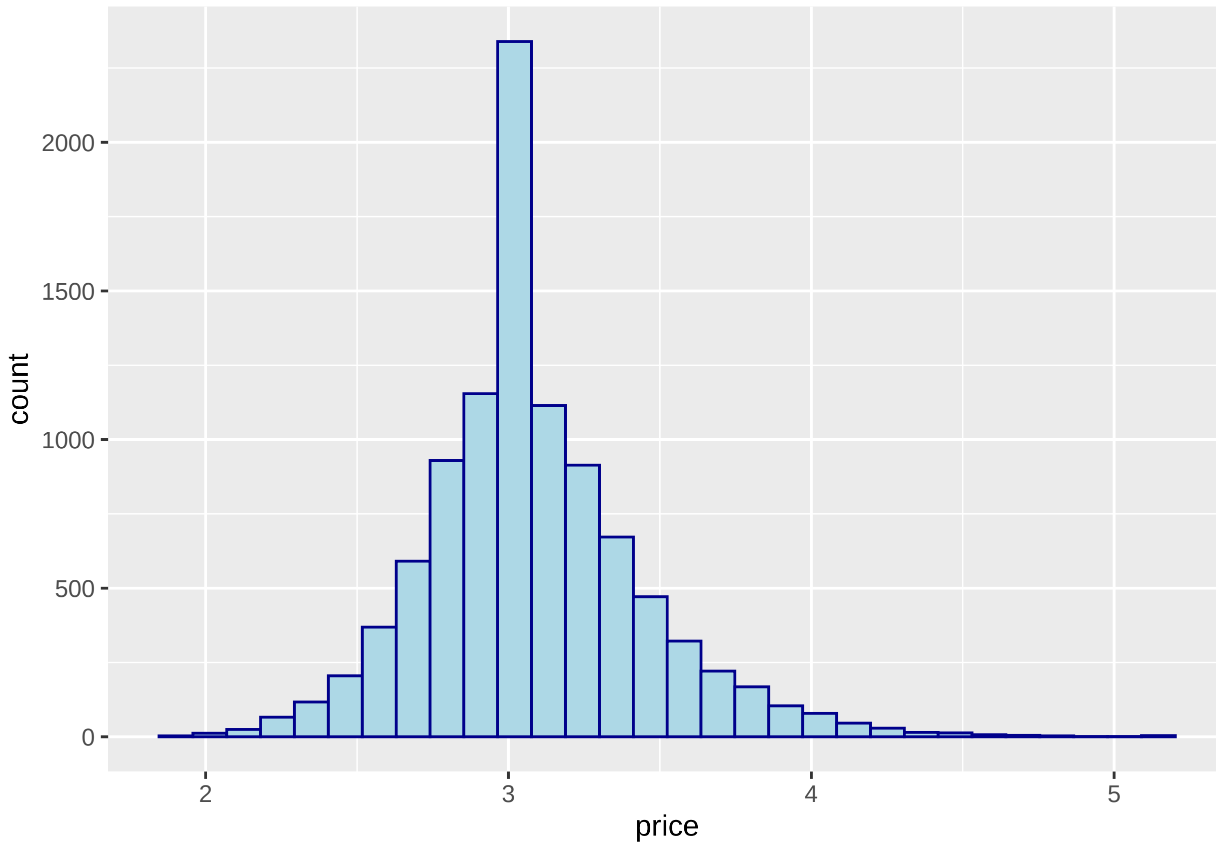
```
# Function for Simulating Brownian Motion & Plotting Histogram
BrownianMotionWithDrift_Histogram<-function(nsim,t,n,mu,sigma)
{

    gbm <- matrix(ncol = nsim, nrow = n)

    for (simu in 1:nsim) {

        gbm[1, simu] <- 0

        for (day in 2:n) {

            gbm[day, simu] <- gbm[(day-1), simu] + rnorm(1,(mu*(t/n)),sqrt(sigma*(t/n)))

        }
    }


    gbm_df <- as.data.frame(gbm) %>%
    mutate(ix = 1:nrow(gbm)) %>%
    pivot_longer(-ix, names_to = 'sim', values_to = 'price')
```

```
    ggplot(data=gbm_df,aes(x=price, color=sim)) +
    geom_histogram(color="darkblue", fill="lightblue",bins=30)+
    theme(legend.position = 'none')

}

# Plotting Histogram
BrownianMotionWithDrift_Histogram(10000,10,10,0,3)
```




```
# Plotting Histogram for Geometric Brownian Motion
GeometricBrownianMotion_Histogram<-function(nsim,t,mu,sigma,S0)
{

    gbm <- matrix(ncol = nsim, nrow = t)

    for (simu in 1:nsim) {

    gbm[1, simu] <- S0
    for (day in 2:t) {

      epsilon <- rnorm(1)
      dt = 1 / 365
      gbm[day, simu] <- gbm[(day-1), simu] * exp((mu - sigma * sigma / 2) * dt + sigma * epsilon * sqrt

        }
    }
```

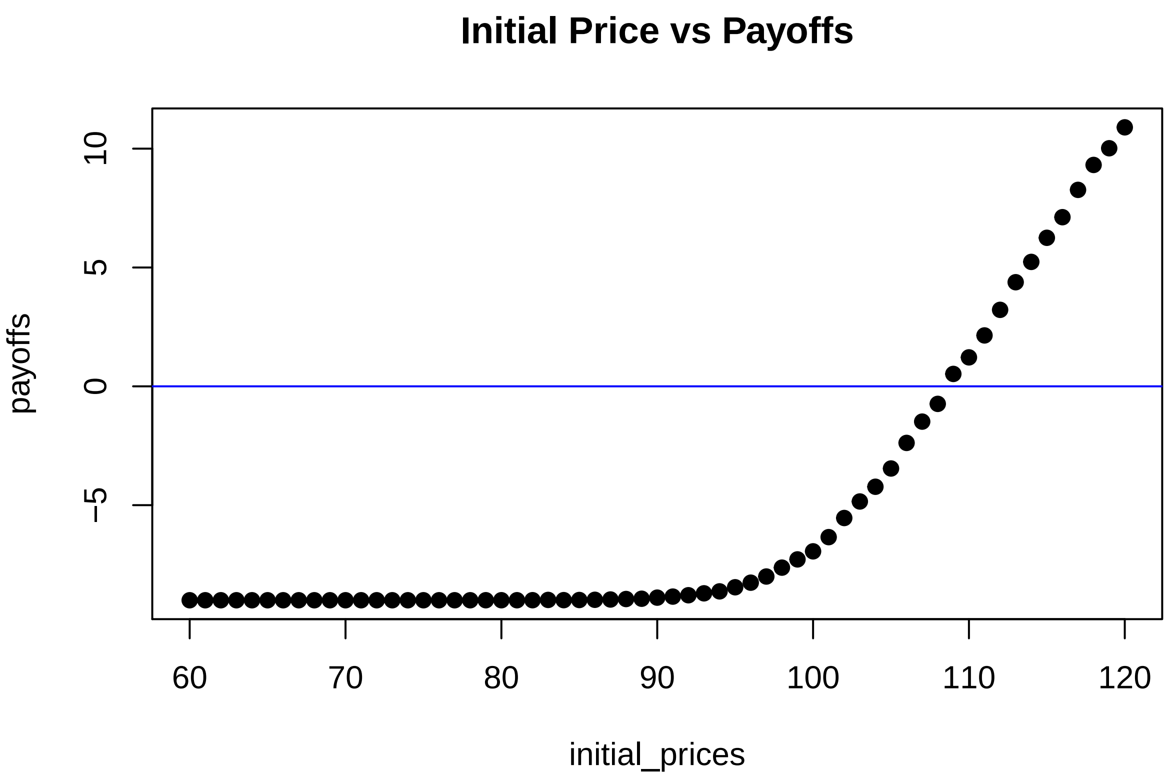
```r
    gbm_df <- as.data.frame(gbm) %>%
    mutate(ix = 1:nrow(gbm)) %>%
    pivot_longer(-ix, names_to = 'sim', values_to = 'price')


    ggplot(gbm_df, aes(x=price))+
    geom_histogram(color="darkblue", fill="lightblue",bins=30)+
    theme(legend.position = 'none')


}

GeometricBrownianMotion_Histogram(1000,10,2,1,3)
```



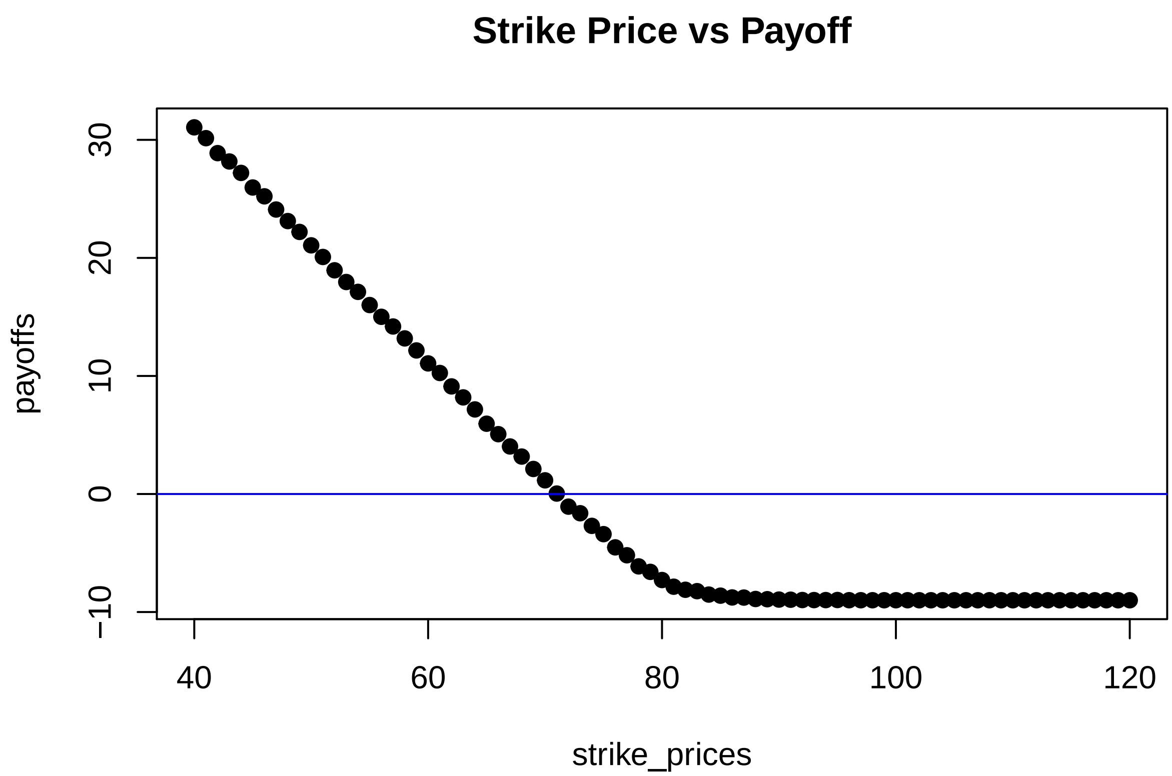
Financial Options & Black Scholes Pricing Model

```r
# Calculates mean expected payoff given initial stock price, premium, expiration...


# Finds Payoff by Simulating Brownian Motion

FinancialOptions<-function(initial_price,premium,expiration_date,strike_price,mu,sigma)
{
```

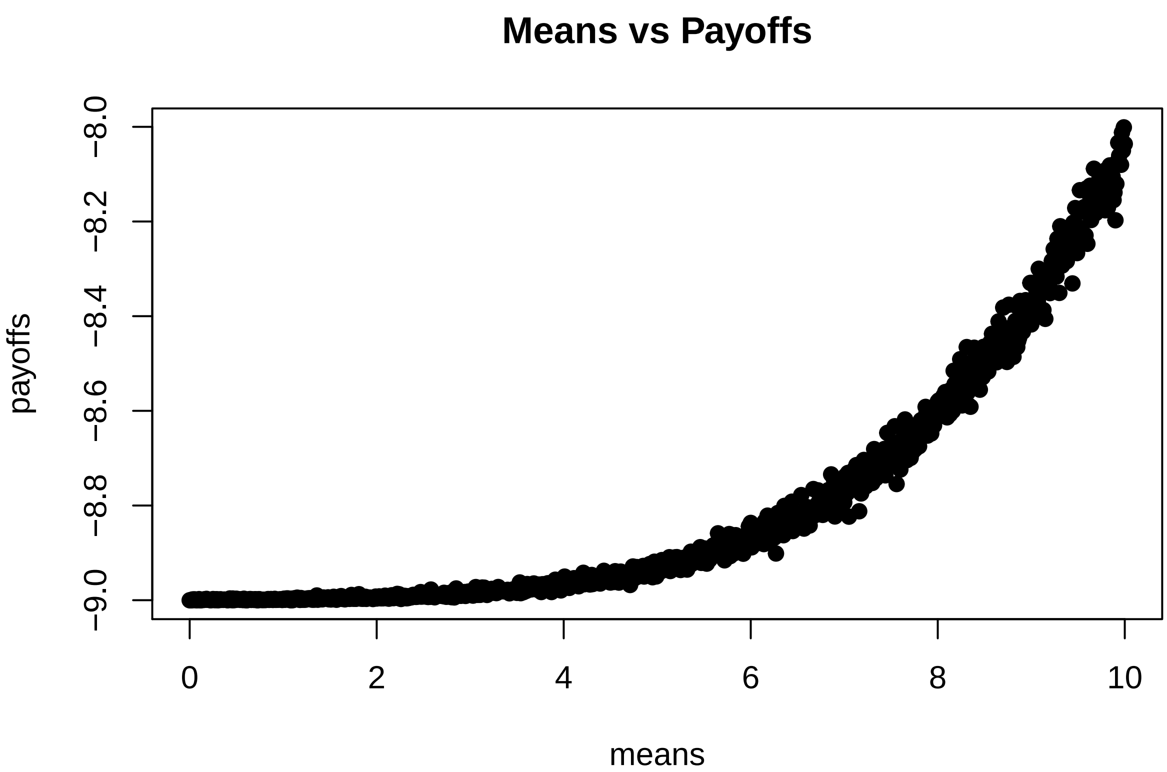
```r
    nsim<-1000
    t<-expiration_date

    gbm <- matrix(ncol = nsim, nrow = t)

    for (simu in 1:nsim) {

    gbm[1, simu] <- initial_price
    for (day in 2:t) {

      epsilon <- rnorm(1)
      dt = 1 / 365
      gbm[day, simu] <- gbm[(day-1), simu] * exp((mu - sigma * sigma / 2) * dt + sigma * epsilon * sqrt

        }
    }

    gbm_df <- as.data.frame(gbm) %>%
    mutate(ix = 1:nrow(gbm)) %>%
    pivot_longer(-ix, names_to = 'sim', values_to = 'price')




    X<-seq(0,length(gbm_df$price))

    for(i in 1:length(gbm_df$price))
    {
        X[i]<-max(gbm_df$price[i]-strike_price,0)
    }

    return(mean(X)-premium)

}

# Simulating Payoffs for Different initial prices

initial_prices<-60:120
payoffs<-c()

for(i in initial_prices){
    payoffs<-c(payoffs,FinancialOptions(i,10,10,100,0.1,0.5))
}

plot(x=initial_prices,y=payoffs,pch=19,main="Initial Price vs Payoffs")
abline(h=0, col="blue")
abline(h=100+10, col="blue")
```



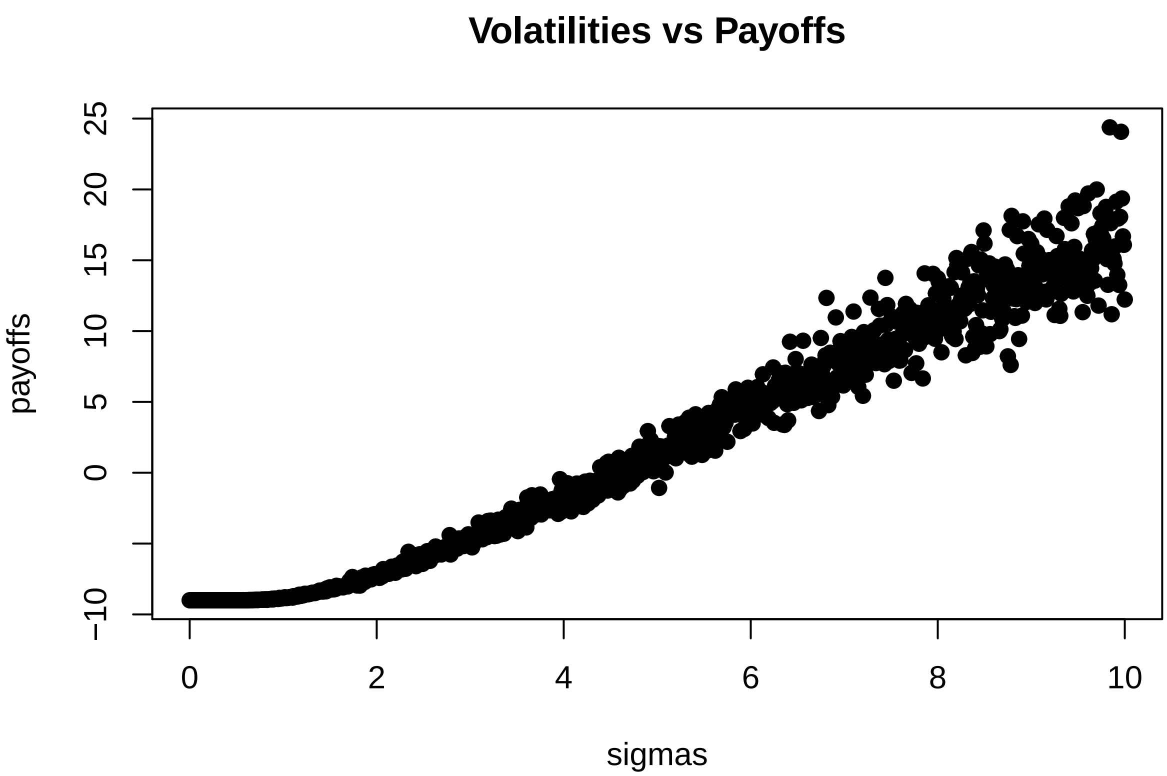

```r
# Simulating Payoffs for Different strike prices
strike_prices<-40:120
payoffs<-c()

for(i in strike_prices){
    payoffs<-c(payoffs,FinancialOptions(80,10,10,i,0.1,0.5))
}

plot(x=strike_prices,y=payoffs,pch=19,main="Strike Price vs Payoff")
abline(h=0, col="blue")
```




```r
#Simulating Payoffs for Different means

means<-seq(0,10,0.01)
payoffs<-c()

for(i in means){
    payoffs<-c(payoffs,FinancialOptions(80,10,10,100,i,0.5))
}

plot(x=means,y=payoffs,pch=19,main="Means vs Payoffs")
```




```r
#Simulating Payoffs for Different volatilities

sigmas<-seq(0,10,0.01)
payoffs<-c()

for(i in sigmas){
    payoffs<-c(payoffs,FinancialOptions(80,10,10,100,0.1,i))
}

plot(x=sigmas,y=payoffs,pch=19,main="Volatilities vs Payoffs")
```




```
# Black Scholes Option Pricing

# Gives Optimal Price to Pay for Given Option

BlackScholesOptionPricing<-function(initial_price,strike_price,expiration_date,interest_rate,volatility)
{
    t<-expiration_date/365

    alpha<-(log(strike_price/initial_price)-(interest_rate-volatility/2)*t)/sqrt(volatility)


    (initial_price*pnorm((alpha-sqrt(volatility)*t)/sqrt(t), mean = 0, sd = 1, lower.tail=FALSE))-
    (exp(-interest_rate*t)*strike_price*pnorm(alpha/sqrt(t),mean=0,sd=1,lower.tail=FALSE))

}

# Optimal price for premium for stock option priced at 80$ with strike price $100, 2% Interest Rate & s
BlackScholesOptionPricing(80,100,90,0.002,0.5)
```

```
## [1] 4.966207
```