Homework 2 Solution

Problem 1: Normal and Lognormal Distributions Let r be the log return and R be the corresponding simple net return, then $r = \log(1 + R)$.

(a) Given that $r \sim N(0.8, 0.5^2)$, we have $1 + R \sim$ lognormal distribution with expected value and variance as

$$E(1+R) = \exp\left(0.8 + \frac{0.5^2}{2}\right) = 2.52$$

$$\text{Var}(1+R) = \exp\left[2 \times 0.8 + 0.5^2\right]\left[\exp(0.5^2) - 1\right] = 1.81$$

Therefore, the expected value and variance for R is

$$E(R) = 2.52 - 1 = 1.52 \qquad \text{Var}(R) = 1.81 \qquad \text{Sd}(R) = \sqrt{1.81} = 1.35$$

(b) R code attached below, sample mean and sample standard deviation for simple net returns are very close to results in part(a).

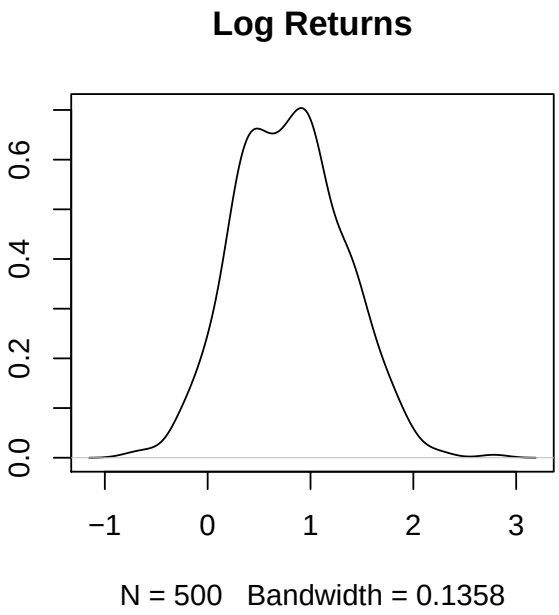
(c) Log returns that are generated from a normal distribution form a empirical density curve that is bell-shaped. The corresponding simple net returns form a empirical density curve that is skewed to the right, just like lognormal distributions.

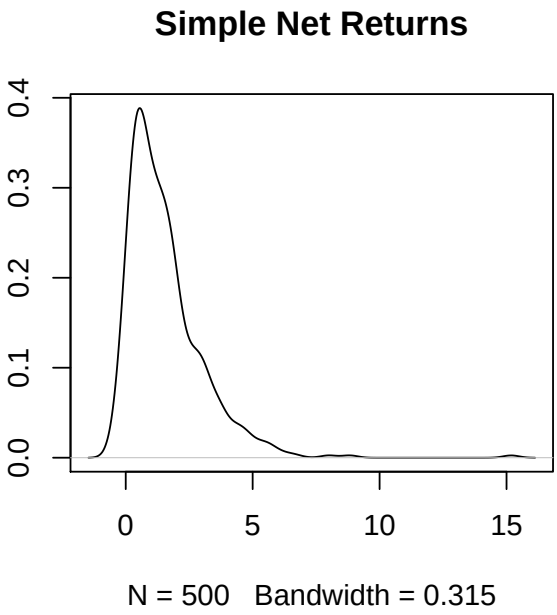
```
#(b)#
logreturn=rnorm(500, mean=0.8, sd=0.5) # simulate random sample
netreturn=exp(logreturn)-1 # calculate corresponding simple net returns
mean(netreturn); sd(netreturn) # close to theoretical values in part(a)
```

```
## [1] 1.553599
```

```
## [1] 1.509747
```

```
#(c)#
par(mfrow=c(1,2))
plot(density(logreturn), main="Log Returns", ylab="")
plot(density(netreturn), main="Simple Net Returns", ylab="")
```





Problem 2.

(a) Given that $p_0 = 12$ and $r_1, \cdots, r_{20} \sim_{iid} N(0.1, 0.5^2)$, since $p_{20} = p_0 + r_1 + \cdots + r_{20}$, we have

$$p_{20} \sim N(12 + 20 \cdot 0.1, 20 \cdot 0.5^2) = \mathbf{N(14, 5)}$$

The probability that $P(p_{20} > 16)$ is computed below using R as **0.1855**.

On the other hand, since $P_{20} = \exp(p_{20})$, P_{20} ∼lognormal distribution with mean and variance as

$$E(P_{20}) = \exp\left(14 + \frac{5}{2}\right) = 14650719$$

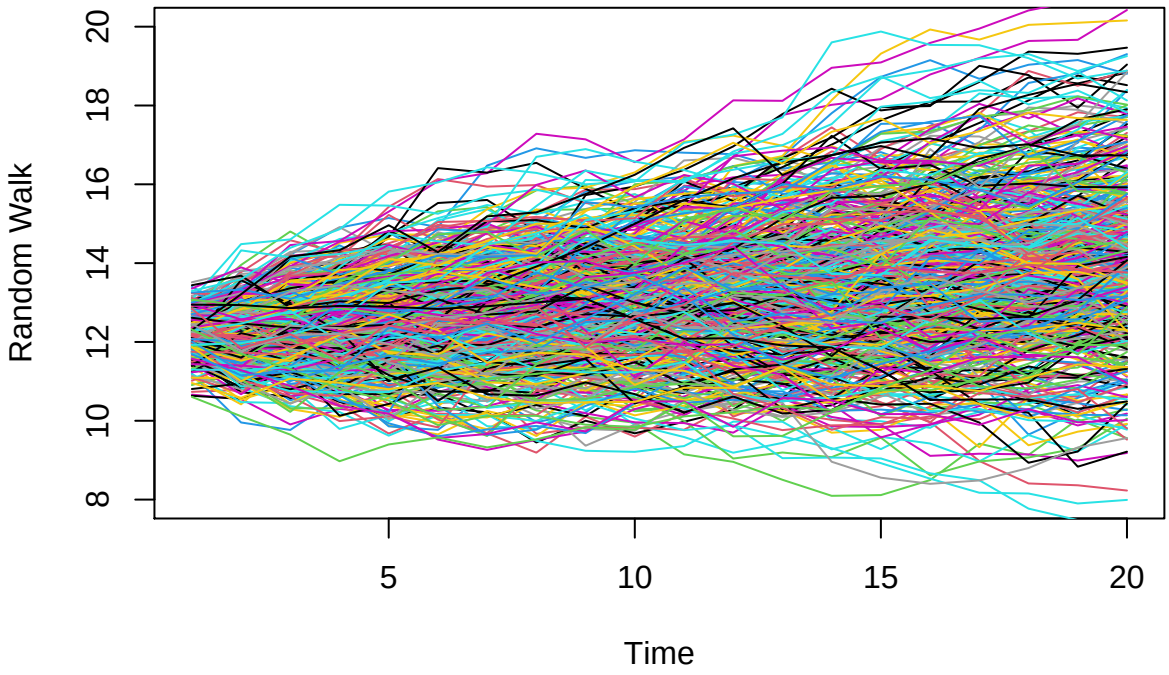
$$\text{Var}(P_{20}) = \exp\left[2 \times 14 + 5\right]\left[\exp(5) - 1\right] = 3.164129e + 16$$

```
#(a)#
pnorm(16, mean=14, sd=sqrt(5), lower.tail = F)
```

```
## [1] 0.1855467
```

```
#(b)#
set.seed(1234)
p0=12 # initial log price
T=20 # total number of steps
sim=500 # number of simulations
rt=rnorm(T*sim, mean=0.1, sd=0.5) # generate T*sim random variables for asset returns
dim(rt)=c(T, sim) # change rt into a T*sim matrix layout
pt=p0+apply(rt, 2, cumsum) # compute cumulative sums for each column (dimension 2)

x=1:T # sequence of time index 1,2, ... , 200
plot(x,pt[,1],xlab="Time",ylab="Random Walk",type='l',ylim=c(8, 20)) # plot 1st column/random walk
for (i in 2:sim){
  lines(x, pt[,i], col=i) # add lines for columns 2-1000
}
```




```
sum(pt[20,]>16)/sim
```

```
## [1] 0.174
```

Problem 3. (a) and (b): see R script below.

(c): Recall that in general a Geometric Brownian Motion process

$$S_t = S_0 \exp\left[(\mu - \frac{\sigma^2}{2})t + \sigma W_t\right]$$

satisfies the following SDE

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t.$$

Here drift $\mu = 0.2$ and volatility $\sigma = 0.5$, given $t = 0.8$, we have

$$S_{0.8} = 100 \exp\left[(0.2 - \frac{0.5^2}{2})0.8 + 0.5W_{0.8}\right] = 100 \exp(0.06 + 0.5W_{0.8}),$$

where $W_{0.8} \sim N(0, 0.8)$. Let $X = 0.06 + 0.5W_{0.8}$, then $X \sim N(0.06, 0.5^2 \times 0.8) = N(0.06, 0.2)$. Now

$$P(S_{0.8} > 120) = P\left(\frac{S_{0.8}}{100} > \frac{120}{100}\right) = P\left(log(\frac{S_{0.8}}{100}) > log(\frac{120}{100})\right) = P\left(X > log(\frac{120}{100})\right) = 0.39227.$$

Moreover, $S_{0.8} \sim$ lognormal distribution with

$$E(S_{0.8}) = 100 \exp(0.2 \times 0.8) = 117.3511$$

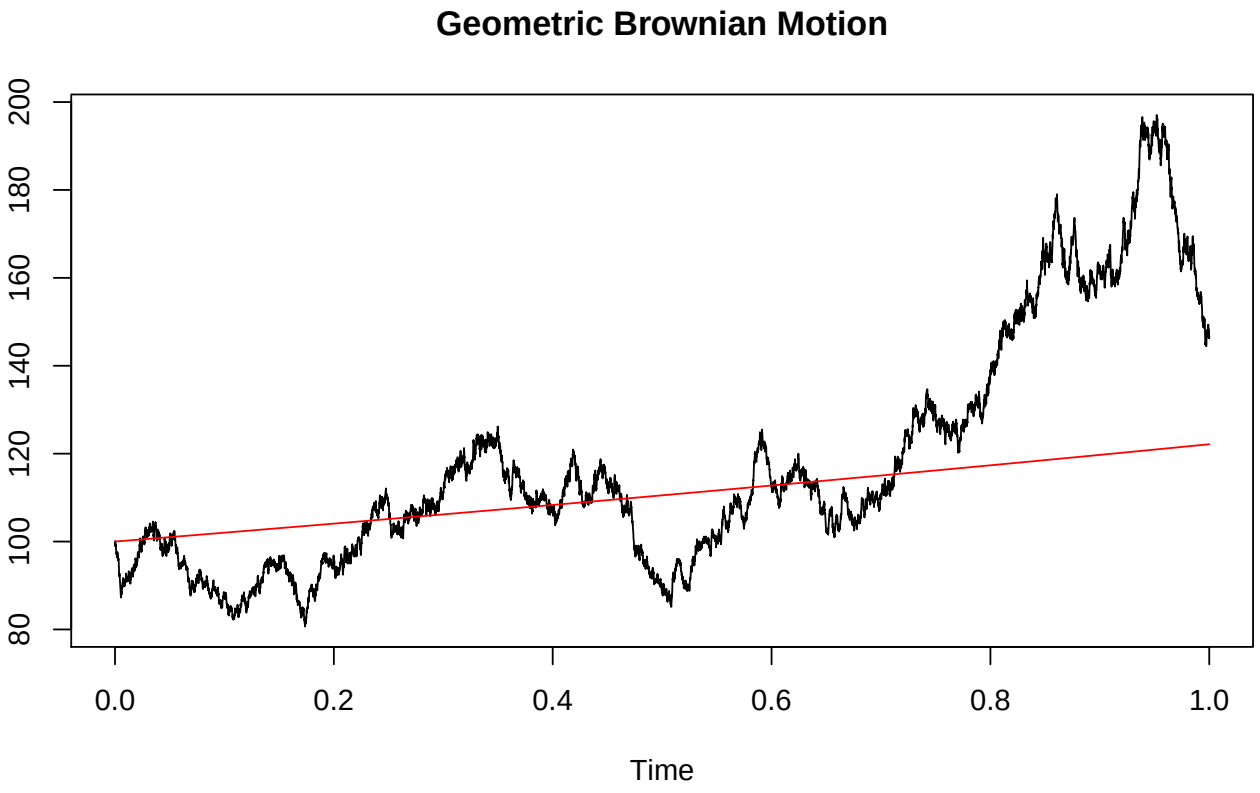
and

$$Var(S_{0.8}) = 100^2 \exp(2 \times 0.2 \times 0.8)[\exp(0.5^2 \times 0.8) - 1] = 3048.999.$$

```
########(a)########
set.seed(1234)
# specify parameters
mu=0.2 # drift
sigma=0.5 # volatility
T=1 # time interval [0,T]
S0=100 # initial price
n=10000 # divide the interval [0,T] into n subintervals
dt=T/n # time length for each subinterval
t=seq(0,T,by=dt) # set up time points t0, t1, t2, ...., tn where ti=i/n for i=0,1, ..., T

# simulate one path
R= mu*dt+sigma*rnorm(n,mean=0,sd=sqrt(dt)) # simulate R.V.s for changes based on RHS of SDE
S=c(S0, rep(0,n)) # initialize vector for stock price process
  for (i in 1:n){
    S[i+1]=S[i]*R[i]+S[i] # calculate stock price at the end of each step
  }
plot(t,S,type="l",main="Geometric Brownian Motion", xlab="Time", ylab="")

mean=c(S0*exp(mu*t)) # mean trend: St=S0*exp(mu*t)
lines(t,mean,col="red") # add red line for mean trend
```




```
########(b)########
# specify parameters
mu=0.2 # drift
```

```r
sigma=0.5 # volatility
T=1 # time interval [0,T]
S0=100 # initial price
n=10000 # divide the interval [0,T] into n subintervals
N=500 # number of paths to be simulated
dt=T/n # time length for each subinterval
t=seq(0,T,by=dt) # set up time points t0, t1, t2, ...., tn where ti=i/n for i=0,1, ..., T

R= mu*dt+sigma*rnorm(n*N,mean=0,sd=sqrt(dt))
R= matrix(R, n, N)

S= matrix(rep(0,n*N), n, N) # initialize matrix for stock price process
S= rbind(rep(S0, N), S) # add starting price for each path
for (j in 1:N){
  for (i in 1:n){
    S[i+1,j]=S[i,j]*R[i,j]+S[i,j]
  }
}

sum(S[8001,]>120)/N
```

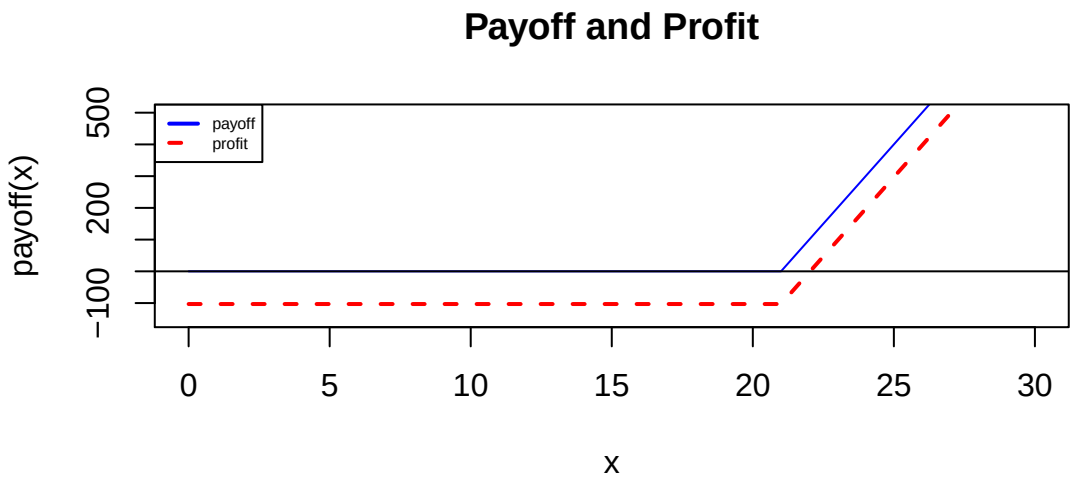
```
## [1] 0.386
```

```r
########(c)########
pnorm(log(120/100), mean=0.06, sd=sqrt(0.2), lower.tail = F)
```

```
## [1] 0.392227
```

Problem 4.

```r
#(a)#
r <- 0.12 # interest rate
premium <- 100 # premium per share
T <- 0.25 # expiration date
K <- 21 # strike price
N=100 # number of shares
payoff <- function(x) sapply(x, function(x) max(c(x - K, 0))*N)
profit <- function(x) sapply(x, function(x) max(c(x - K, 0))*N - premium*exp(r*T))
curve(payoff, 0, 30, main = "Payoff and Profit", col = "blue", lty=1,lwd=1, ylim=c(-150,500))
curve(profit, 0, 30, col ="red", add = TRUE, lty = 2, lwd = 2)
abline(h=0)
legend("topleft", c("payoff", "profit"), lty = c(1, 2), col = c("blue", "red"), lwd=c(2,2), cex = 0.5)
```




```r
#(b)#
payoff <- function(x) sapply(x, function(x) -max(c(x - K, 0))*N*exp(-r*T))
profit <- function(x) sapply(x, function(x) -(max(c(x - K, 0))*N*exp(-r*T) - premium))
curve(payoff, 0, 30, main = "Payoff and Profit", col = "blue", lty=1,lwd=1, ylim=c(-500,150))
curve(profit, 0, 30, col ="red", add = TRUE, lty = 2, lwd = 2)
abline(h=0)
legend("bottomleft", c("payoff", "profit"), lty = c(1, 2), col = c("blue", "red"), lwd=c(2,2), cex = 0.5
```