2022년 2학기

Computational Statistics

HW#9

222STG10

김희숙

Problem

교재 6.6, Asian option MC 예제, 6.7 을 풀어라.

6.6. Consider testing the hypotheses $H_0 : \lambda = 2$ versus $H_a : \lambda > 2$ using 25 observations from a Poisson(λ) model. Rote application of the central limit theorem would suggest rejecting H_0 at $\alpha = 0.05$ when $Z \geq 1.645$, where $Z = (\bar{X} - 2)/\sqrt{2/25}$.

a. Estimate the size of this test (i.e., the type I error rate) using five Monte Carlo approaches: standard, antithetic, importance sampling with unstandardized and standardized weights, and importance sampling with a control variate as in Example 6.12. Provide a confidence interval for each estimate. Discuss the relative merits of each variance reduction technique, and compare the importance sampling strategies with each other. For the importance sampling approaches, use a Poisson envelope with mean equal to the H_0 rejection threshold, namely $\lambda = 2.4653$.

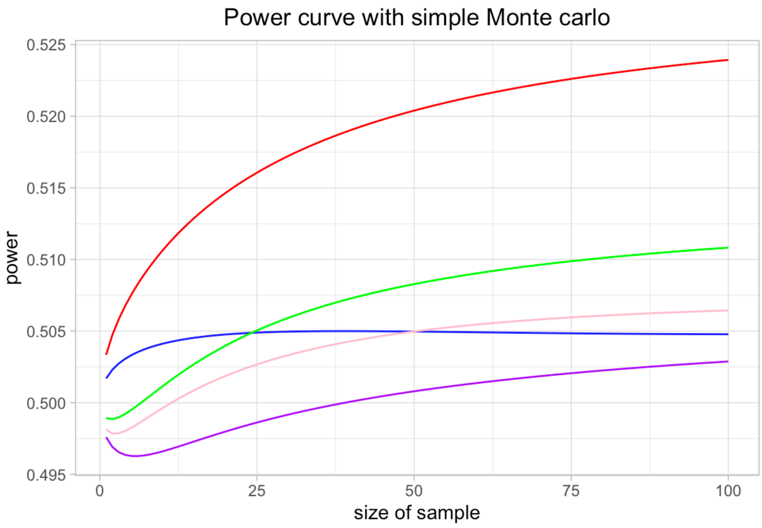
b. Draw the power curve for this test for $\lambda \in [2.2, 4]$, using the same five techniques. Provide pointwise confidence bands in each case. Discuss the relative merits of each technique in this setting. Compare the performances of the importance sampling strategies with their performance in part (a).

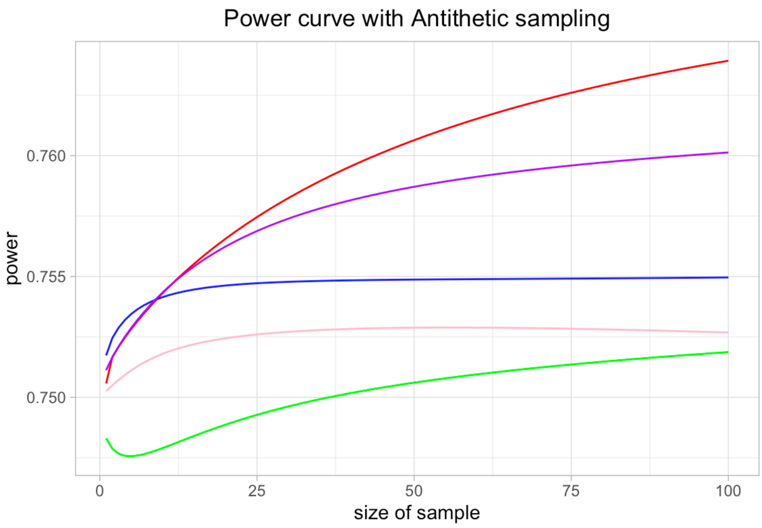
a.

Size of the test				
Naive MC	Antithetic sampling	Importance sampling with unstandardized weight	Importance sampling with standardized weight	control variate
0.057	0.029	0.4930537	0.4927981	0.4930789

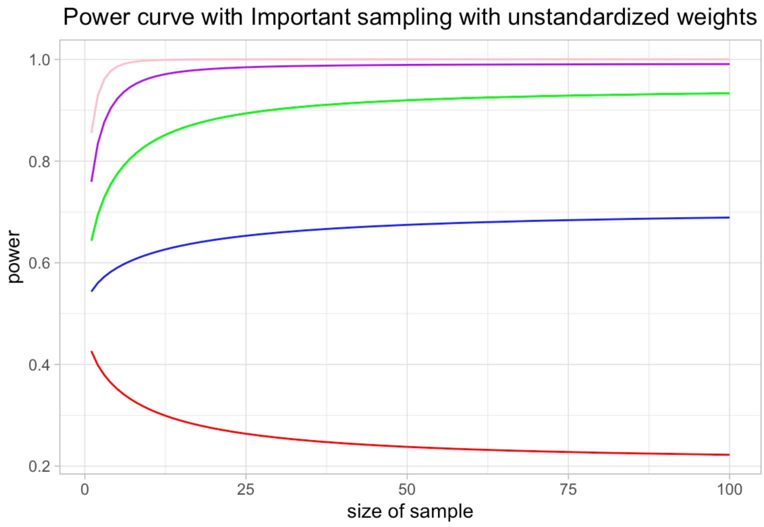
Confidence Interval for alpha=0.05		
name	**lower**	**upper**
Simple monte carlo	0.05534495	0.05626105
Antithetic Sampling	0.02745848	0.02808552
Important sampling with unstandardized weights	0.4965016	0.4983161
Important sampling with standardized weights	0.4964278	0.4983295
Important sampling with control variate	0.4965784	0.4983929

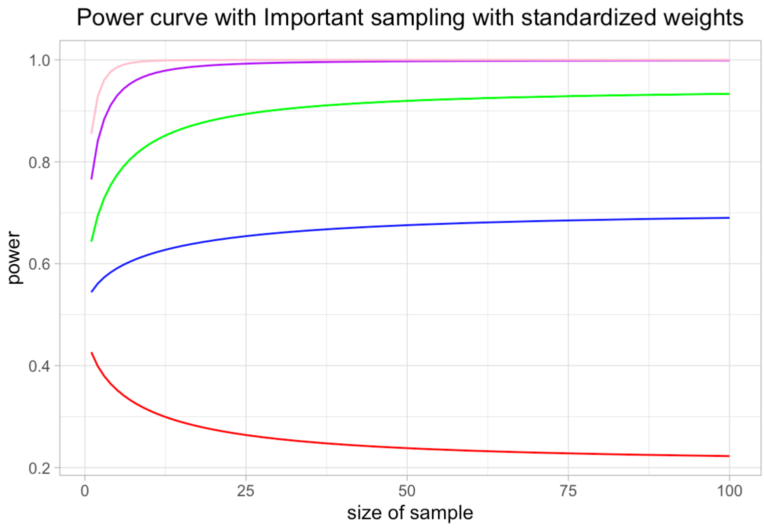
b.

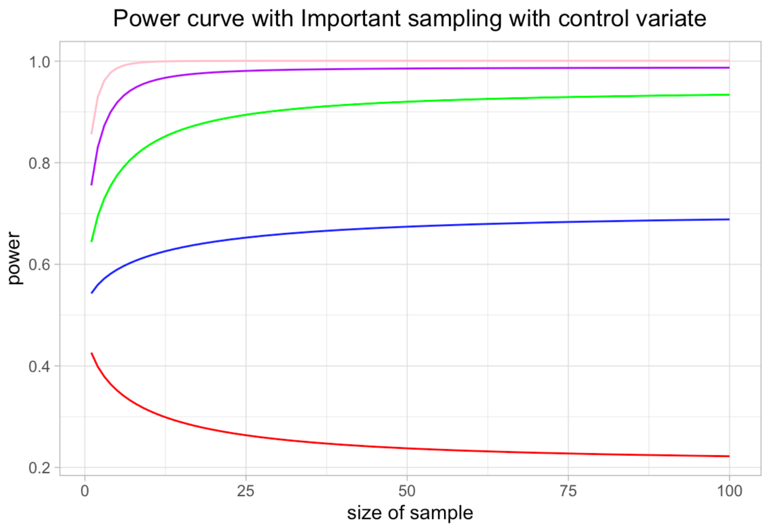




두 그래프의 red라인과 blue라인은 유사해보이지만, 다른 3개의 라인은 두 그래프에서 큰 차이를 보인다.







세 그래프가 모두 유사한 것을 확인 할 수 있다.

Asian option MC

mu_mc	mu_geom	mu_cv	sd(mc)	sd(cv)
1.87721	1.82778	1.839492	0.1102977	0.1079317

	Min	1st Qu	Median	Mean	3rd Qu	Max
mc	1.523	1.802	1.871	1.877	1.951	2.170
cv	1.554	1.766	1.844	1.838	1.911	2.184

6.7. Consider pricing a European call option on an underlying stock with current price $S^{(0)} = 50$, strike price $K = 52$, and volatility $\sigma = 0.5$. Suppose that there are $N = 30$ days to maturity and that the risk-free rate of return is $r = 0.05$.

a. Confirm that the fair price for this option is 2.10 when the payoff is based on $S^{(30)}$ [i.e., a standard option with payoff as in (6.74)].

b. Consider the analogous Asian option (same $S^{(0)}$, K, σ, N, and r) with payoff based on the arithmetic mean stock price during the holding period, as in (6.77). Using simple Monte Carlo, estimate the fair price for this option.

c. Improve upon the estimate in (b) using the control variate strategy described in Example 6.13.

d. Try an antithetic approach to estimate the fair price for the option described in part (b).

e. Using simulation and/or analysis, compare the sampling distributions of the estimators in (b), (c), and (d).

a.

Mean은 2.184705 이다.

b.

Monte carlo를 통해 구한 mean은 0.9348154 이다.

c.

Control variate를 통해 구한 mean은 0.9059664 이다.

d.

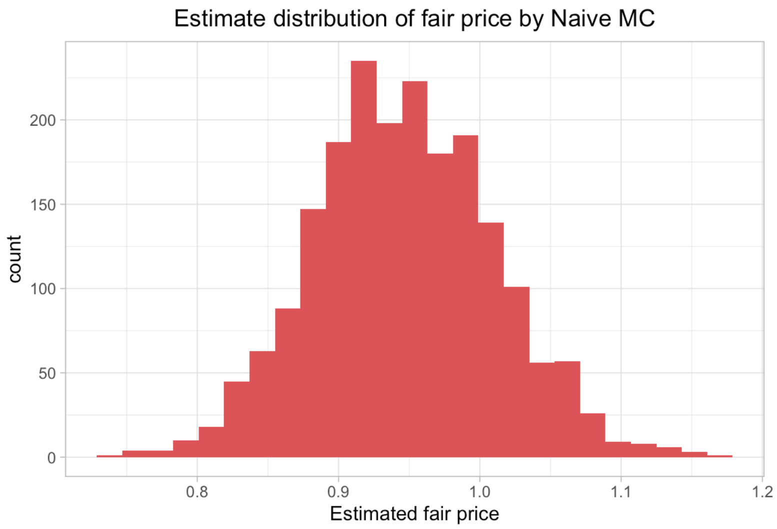
Antithetic approach를 통해 구한 mean은 0.8971104 이다. 앞서(b,c)를 통해 구한 값과 비교해보면 상당히 작은 것을 확인 할 수 있다.

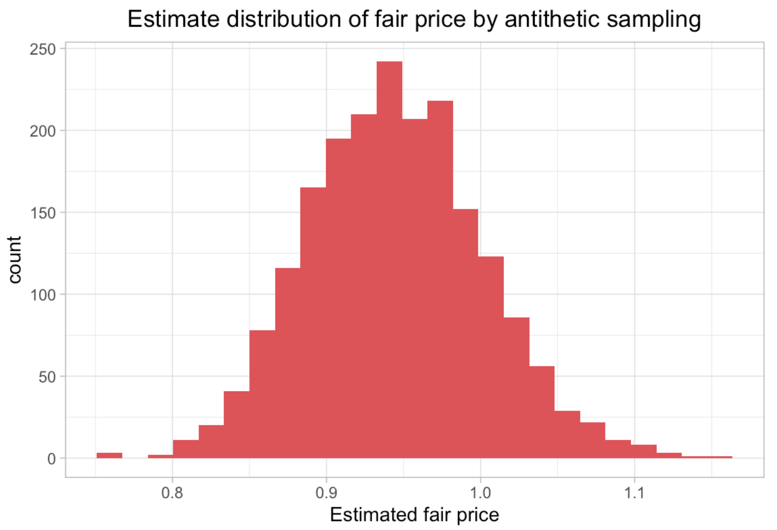
e.

Monte carlo, Control variate, Antithetic approach 방식으로 구한 값들의 표준편차를 구해보면 다음과 같다. 세 방식으로 구한 값의 표준편차가 유사한 것을 알 수 있다.

sd(mc)	sd(cv)	sd(as)
0.06400576	0.06205592	0.05635452

	Min	1st Qu	Median	Mean	3rd Qu	Max
mc	0.7386	0.9032	0.9451	0.9462	0.9898	1.1706
cv	0.6790	0.8535	0.8969	0.8957	0.9374	1.0961
as	0.7572	0.9054	0.9438	0.9448	0.9812	1.1534





각 방식을 통해 추정한 값들의 분포가 유사한 것을 볼 수 있다.

Code appendix

```
# 6.6
### a
X <- list()
for (i in 1:1000) {
  X[[i]] <- rpois(25, 2)
}

get_hx <- function(X) {
  h <- c()
  for (n in 1:length(X)) {
    xi <- X[[n]]
    z <- (mean(xi)-2)/sqrt(2/25)
    h <- append(h, ifelse(z>=1.645, 1, 0))
  }
  return(h)
}
# Naive MC
a_mc <- mean(get_hx(X))
a_mc
# Antithetic sampling
a_as <- sum(get_hx(X[1:500])+get_hx(lapply(X[1:500], function(x) -x)))/1000
a_as
# data from envelope function
X_env <- list()
for (i in 1:1000) {
  X_env[[i]] <- rpois(25, 2.4653)
}

# Importance sampling with unstandardized weight
w <- c()
for (i in 1:length(X_env)) {
  w <- append(w, dpois(X_env[[i]], 2)/dpois(X_env[[i]], 2.4653))
}
ISUW <- mean(get_hx(X_env)*w)
ISUW
# Importance sampling with standardized weight
w <- c()
for (i in 1:length(X_env)) {
  w <- append(w, dpois(X_env[[i]], 2)/dpois(X_env[[i]], 2.4653))
```

```
}
ISSW <- sum(get_hx(X_env)*w/sum(w))
ISSW
# control variate
y <- get_hx(X)*w
lambda <- lm(y~w)$coefficients[2]
ISCV <- ISUW + lambda*(mean(w)-1)
ISCV
## replicating simulation function
simulation <- function(lambda) {
  # 결과값을 저장할 빈 벡터 생성
  mc <- c()
  as <- c()
  ISUW <- c()
  ISSW <- c()
  ISCV <- c()


  # 1000개의 estimate 생성
  for (ii in 1:1000) {

    # random data 생성
    X <- list()
    for (i in 1:1000) {
      X[[i]] <- rpois(25, lambda)
    }

    # random data for envelope function
    X_env <- list()
    for (i in 1:1000) {
      X_env[[i]] <- rpois(25, 2.4653)
    }
w <- c()   # weight
    for (i in 1:length(X_env)) {
      w <- append(w, dpois(X_env[[i]], lambda)/dpois(X_env[[i]], 2.4653))
    }

    # calculate estimate
    mc <- append(mc, mean(get_hx(X)))
    as <- append(as, sum(get_hx(X[1:500])+get_hx(lapply(X[1:500], function(x) -x)))/1000)
```

```
    ISUW <- append(ISUW, mean(get_hx(X_env)*w))
    ISSW <- append(ISSW, sum(get_hx(X_env)*w/sum(w)))

    y <- get_hx(X)*w
    pois_l <- lm(y~w)$coefficients[2]
    ISCV <- ISUW + pois_l*(mean(w)-1)
  }
  return(data.frame(mc, as, ISUW, ISSW, ISCV))
}
result_2 <- simulation(2)
head(result_2)
cat("Confidence Interval for alpha=0.05", "\n","\n",
    "[Simple monte carlo]", "\n",
    "lower:", mean(result_2$mc)-qnorm(1-0.05/2)*sd(result_2$mc)/sqrt(nrow(result_2)),
    "upper:", mean(result_2$mc)+qnorm(1-0.05/2)*sd(result_2$mc)/sqrt(nrow(result_2)), "\n","\n",
    "[Antithetic Sampling]", "\n",
    "lower:", mean(result_2$as)-qnorm(1-0.05/2)*sd(result_2$as)/sqrt(nrow(result_2)),
    "upper:", mean(result_2$as)+qnorm(1-0.05/2)*sd(result_2$as)/sqrt(nrow(result_2)), "\n","\n",
    "[Important sampling with unstandardized weights]", "\n",
    "lower:", mean(result_2$ISUW)-qnorm(1-0.05/2)*sd(result_2$ISUW)/sqrt(nrow(result_2)),
    "upper:",       mean(result_2$ISUW)+qnorm(1-0.05/2)*sd(result_2$ISUW)/sqrt(nrow(result_2)),
"\n","\n",
    "[Important sampling with standardized weights]", "\n",
    "lower:", mean(result_2$ISSW)-qnorm(1-0.05/2)*sd(result_2$ISSW)/sqrt(nrow(result_2)),
    "upper:",       mean(result_2$ISSW)+qnorm(1-0.05/2)*sd(result_2$ISSW)/sqrt(nrow(result_2)),
"\n","\n",
    "[Important sampling with control variate]", "\n",
    "lower:", mean(result_2$ISCV)-qnorm(1-0.05/2)*sd(result_2$ISCV)/sqrt(nrow(result_2)),
    "upper:", mean(result_2$ISCV)+qnorm(1-0.05/2)*sd(result_2$ISCV)/sqrt(nrow(result_2)), "\n")


### b
get_power <- function(X, lambda, n) {
  h <- c()
  for (l in 1:length(X)) {
    xi <- X[[l]]
    beta <- pnorm((mean(xi)-lambda)/sqrt(2/n))
    h <- append(h, 1-beta)
  }
  return(h)
}
power <- function(lambda){
  X <- list()
```

```
  for (i in 1:1000) {
    X[[i]] <- rpois(25, lambda)
  }

  X_env <- list()
  for (i in 1:1000) {
    X_env[[i]] <- rpois(25, 2.4653)
  }

  w <- c()   # weight
  for (i in 1:length(X_env)) {
    w <- append(w, dpois(X_env[[i]], lambda)/dpois(X_env[[i]], 2.4653))
  }

  mc <- c()
  as <- c()
  ISUW <- c()
  ISSW <- c()
  ISCV <- c()
for (n in 1:100) {
    mc <- append(mc, mean(get_power(X, lambda, n)))
    as <- append(as, sum(get_power(X[1:500], lambda, n)+get_power(lapply(X[1:500], function(x)
-x), lambda, n))/1000)
    ISUW <- append(ISUW, mean(get_power(X_env, lambda, n)*w))
    ISSW <- append(ISSW, sum(get_power(X_env, lambda, n)*w/sum(w)))

    y <- get_power(X, lambda, n)*w
    pois_l <- lm(y~w)$coefficients[2]
    ISCV <- ISUW + pois_l*(mean(w)-1)
  }
  return(data.frame(mc, as, ISUW, ISSW, ISCV))
}
df.22 <- power(lambda=2.2)
df.26 <- power(lambda=2.6)
df.30 <- power(lambda=3)
df.35 <- power(lambda=3.5)
df.40 <- power(lambda=4)
# Power curve with simple Monte carlo
data.frame(n=1:100, n.22=df.22$mc, n.26=df.26$mc, n.30=df.30$mc,
         n.35=df.35$mc, n.40=df.40$mc) %>%
  ggplot()+geom_line(aes(n, n.22), color="red")+geom_line(aes(n, n.26), color="blue")+
```

```
  geom_line(aes(n, n.30), color="green")+
  geom_line(aes(n, n.35), color="purple")+geom_line(aes(n, n.40), color="pink")+
  theme_light()+labs(x="size of sample", y="power", title="Power curve with simple Monte
carlo")+
  theme(plot.title = element_text(hjust=0.5))


# Power curve with simple Monte carlo
data.frame(n=1:100, n.22=df.22$as, n.26=df.26$as, n.30=df.30$as,
         n.35=df.35$as, n.40=df.40$as) %>%
  ggplot()+geom_line(aes(n, n.22), color="red")+geom_line(aes(n, n.26), color="blue")+
  geom_line(aes(n, n.30), color="green")+
  geom_line(aes(n, n.35), color="purple")+geom_line(aes(n, n.40), color="pink")+
  theme_light()+labs(x="size of sample", y="power", title="Power curve with Antithetic
sampling")+
  theme(plot.title = element_text(hjust=0.5))


# Power curve with Important sampling with unstandardized weights
data.frame(n=1:100, n.22=df.22$ISUW, n.26=df.26$ISUW, n.30=df.30$ISUW,
         n.35=df.35$ISUW, n.40=df.40$ISUW) %>%
  ggplot()+geom_line(aes(n, n.22), color="red")+geom_line(aes(n, n.26), color="blue")+
  geom_line(aes(n, n.30), color="green")+
  geom_line(aes(n, n.35), color="purple")+geom_line(aes(n, n.40), color="pink")+
  theme_light()+labs(x="size of sample", y="power", title="Power curve with Important sampling
with unstandardized weights")+
  theme(plot.title = element_text(hjust=0.5))


# Power curve with Important sampling with standardized weights
data.frame(n=1:100, n.22=df.22$ISSW, n.26=df.26$ISSW, n.30=df.30$ISSW,
         n.35=df.35$ISSW, n.40=df.40$ISSW) %>%
  ggplot()+geom_line(aes(n, n.22), color="red")+geom_line(aes(n, n.26), color="blue")+
  geom_line(aes(n, n.30), color="green")+
  geom_line(aes(n, n.35), color="purple")+geom_line(aes(n, n.40), color="pink")+
  theme_light()+labs(x="size of sample", y="power", title="Power curve with Important sampling
with standardized weights")+
  theme(plot.title = element_text(hjust=0.5))


  # Power curve with Important sampling with control variate
  data.frame(n=1:100, n.22=df.22$ISCV, n.26=df.26$ISCV, n.30=df.30$ISCV,
           n.35=df.35$ISCV, n.40=df.40$ISCV) %>%
    ggplot()+geom_line(aes(n, n.22), color="red")+geom_line(aes(n, n.26), color="blue")+
    geom_line(aes(n, n.30), color="green")+
    geom_line(aes(n, n.35), color="purple")+geom_line(aes(n, n.40), color="pink")+
    theme_light()+labs(x="size of sample", y="power", title="Power curve with Important sampling
```

```
with control variate")+
   theme(plot.title = element_text(hjust=0.5))



# 6.7
### a
s0 <- 50
K <- 52
vol <- 0.5
N <- 30
r <- 0.05

C <- c()
for (i in 1:1000) {
  s <- s0*exp((r-vol^2/2)*N/365+vol*rnorm(1)*sqrt(N/365))
  C <- append(C, exp(-r*N/365)*max(0, s-K))
}
mean(C)

### b
X <- list()
for (i in 1:1000) {
  X[[i]] <- rnorm(30)
}

get_A <- function(X) {
  A <- c()
  for (n in 1:length(X)) {
    s <- c()
    s_t <- s0
    z <- X[[n]]
    for (i in 1:30) {
      s_t1 <- s_t * exp((r-vol^2/2)/365+vol*z[i]/sqrt(365))
      s <- append(s, s_t1)
      s_t <- s_t1
    }
    sbar <- mean(s)
    A <- append(A, exp(-r*N/365)*max(0, sbar-K))
  }
  return(A)
}
```

```
mu_mc <- mean(get_A(X))
mu_mc

### c
n <- 1000 #the number of prices in the average
c3 <- 1 + 1/n
c2 <- vol * (c3*N/1095*(1+1/(2*n)))^(1/2)
c1 <- (log(s0/K)+c3*N/730*(r-vol^2/2)+c3*vol^2*N/1095*(1+1/(2*n)))/c2

theta <- s0*pnorm(c1)*exp(-N*(r+c3*vol^2/6)*(1-1/n)/730)-
  K*pnorm(c1-c2)*exp(-r*N/365)

get_A_by_geomean <- function(X) {
  A <- c()
  for (n in 1:length(X)) {
    s <- c()
    s_t <- s0
    z <- X[[n]]
    for (i in 1:30) {
      s_t1 <- s_t * exp((r-vol^2/2)/365+vol*z[i]/sqrt(365))
      s <- append(s, s_t1)
      s_t <- s_t1
    }
    sbar <- (prod(s))^(1/N)
    A <- append(A, exp(-r*N/365)*max(0, sbar-K))
  }
  return(A)
}

theta_mc <- mean(get_A_by_geomean(X))

mu_cv <- mu_mc-1*(theta_mc - theta)
mu_cv

### d
mu_as <- sum(get_A(X[1:500])+get_A(lapply(X[1:500], function(x) -x)))/1000
mu_as

### e
mc <- c()
cv <- c()
```

```
as <- c()

for (ii in 1:1000) {
  X <- list()
  for (i in 1:1000) {
    X[[i]] <- rnorm(30)
  }
  mc <- append(mc, mean(get_A(X)))
  cv <- append(cv, mu_mc-1*(mean(get_A_by_geomean(X)) - theta))
  as <- append(as, sum(get_A(X[1:500])+get_A(lapply(X[1:500], function(x) -x)))/1000)
}

sd(mc) ; sd(cv) ; sd(as)
summary(mc)
summary(cv)
summary(as)

df <- data.frame(mc=mc, as=as, cv=cv)

# distribution of naive MC
df %>% ggplot(aes(mc))+geom_histogram(fill="indianred", bins=25)+
  theme_light()+labs(title="Estimate distribution of fair price by Naive MC",
                   x="Estimated fair price") +
  theme(plot.title = element_text(hjust=0.5))
# distribution of antithetic sampling
df %>% ggplot(aes(as))+geom_histogram(fill="indianred", bins=25)+
  theme_light()+labs(title="Estimate distribution of fair price by antithetic sampling",
                   x="Estimated fair price") +
  theme(plot.title = element_text(hjust=0.5))
# distribution of control variate strategy
df %>% ggplot(aes(cv))+geom_histogram(fill="indianred", bins=25)+
  theme_light()+labs(title="Estimate distribution of fair price by control variate strategy",
                   x="Estimated fair price") +
  theme(plot.title = element_text(hjust=0.5))



# Asian option MC
s0 <- 100
K <- 102
N <- 50
vol <- 0.3
```

```
r <- 0.05

X <- list()
for (i in 1:1000) {
  X[[i]] <- rnorm(N)
}

myA <- function(X) {
  A <- c()
  for (n in 1:length(X)) {
    s <- c()
    s_t <- s0
    z <- X[[n]]
    for (i in 1:N) {
      s_t1 <- s_t * exp((r-vol^2/2)/365+vol*z[i]/sqrt(365))
      s <- append(s, s_t1)
      s_t <- s_t1
    }
    sbar <- mean(s)
    A <- append(A, exp(-r*N/365)*max(0, sbar-K))
  }
  return(A)
}

mu_mc <- mean(myA(X))
mu_mc

mygeomA <- function(X) {
  A <- c()
  for (n in 1:length(X)) {
    s <- c()
    s_t <- s0
    z <- X[[n]]
    for (i in 1:N) {
      s_t1 <- s_t * exp((r-vol^2/2)/365+vol*z[i]/sqrt(365))
      s <- append(s, s_t1)
      s_t <- s_t1
    }
    sbar <- (prod(s))^(1/N)
    A <- append(A, exp(-r*N/365)*max(0, sbar-K))
  }
```

```
  return(A)
}

mu_geom <- mean(mygeomA(X))
mu_geom

n <- 1000 #the number of prices in the average
c3 <- 1 + 1/n
c2 <- vol * (c3*N/1095*(1+1/(2*n)))^(1/2)
c1 <- (log(s0/K)+c3*N/730*(r-vol^2/2)+c3*vol^2*N/1095*(1+1/(2*n)))/c2

theta <- s0*pnorm(c1)*exp(-N*(r+c3*vol^2/6)*(1-1/n)/730)-
  K*pnorm(c1-c2)*exp(-r*N/365)

mu_cv <- mu_mc-1*(mu_geom - theta)
mu_cv

mc <- c()
cv <- c()
for (ii in 1:1000) {
  X <- list()
  for (i in 1:1000) {
    X[[i]] <- rnorm(N)
  }
  mc <- append(mc, mean(myA(X)))
  cv <- append(cv, mu_mc-1*(mean(mygeomA(X)) - theta))

}

sd(mc) ; sd(cv)
```