2019statcacul

HolaTeo

install.packages(“foreach”) install.packages(“tidyverse”)

#Q1 ##

```
coin <- function(prob=0.3, bet=1, Pe = 6, Pa = 4){
 rec <- tibble(a = Pe, b = Pa)
 while(all(rec>0)){
    u <- runif(1)
   Pe <- ifelse(u<prob, Pe+bet,Pe-bet)
   Pa <- ifelse(u>=prob, Pa+bet,Pa-bet)
   rec <- rec %>% bind_rows(c(a=Pe, b=Pa))
   if(Pe==0 | Pa==0) break()
 }
 rec %>% mutate(idx = 1:n())  %>%
   select(a, b) %>%
   summarise(Gamesu=n(), who_win=ifelse(last(a)==0, "Paul Win!","Peter Win!"), PeterMoney=mean(a))
}

(resault <- coin(prob=0.3, bet=1, Pe=6, Pa=4))
```

```
## # A tibble: 1 x 3
##   Gamesu who_win    PeterMoney
##    <int> <chr>           <dbl>
## 1     11 Paul Win!        3.73
```

0.1

```
rep <- foreach(i=1:1000, .combine = bind_rows) %do%
               {
                 coin()
} %>%
  summarise(Peter_Win_Prob=mean(who_win=="Peter Win!"),Game_Time=mean(Gamesu),Peter_Money_Mean=mean(Pete

(resault2 <- rep)
```

```
## # A tibble: 1 x 3
##   Peter_Win_Prob Game_Time Peter_Money_Mean
##            <dbl>     <dbl>            <dbl>
## 1          0.028      15.1             3.73
```

#Q2

project 6.B correlation : pearson corr & spearman H0 : rho=0 이 참일 때 rho.hat들의 분포를 찾음

$Corr($ $X_1, ..., X_n$, sample(c($Y_1, ..., Y_n$)))

library(mvtnorm)

n<-20 m<-1000 rho0 <- 0 rho <- c(seq(-1,1,length=40)) #alternative M <- length(rho) power1 <- power2 <- power3 <-

rhotest <- cor.test(samp[,1], samp[,2], method=“pearson”, rhotest$p.value) < -cor.test(samp[,1], samp[,2], method = kendal, rhotest$p.value) plot(rho, power1, type="l") lines(rho, power2, lty=1) lines(rho, power3, lty=2)

#Q3

0.2

```
m<-10000
mc <- function(m){
x <-rcauchy(m)
fg <- as.numeric(x>2)
a1 <- mean(fg)
return(a1)
}
mc(m)
```

```
## [1] 0.1547
```

##important sampling

```
mc.impo1 <- function(m){
x<-2/runif(m)
fg <- 1/(2*(pi*(1+1/x^2)))
b1 <- mean(fg)
return(b1)
}
mc.impo1(m)
```

```
## [1] 0.1475518
```

0.2.1 important sampling2

```
mc.impo2 <- function(m){
x<-runif(m, 0, 0.5)
fg <- dcauchy(x)/2
b2<-mean(fg)
return(b2)
}
mc.impo2(m)
```

```
## [1] 0.1476843
```

##hit or miss

```
mc.hit <- function(m){
x <- runif(m, 0, 2)
y <- runif(m, min=0, max=dcauchy(0))
est <- 1/2 - sum(as.numeric(y<=dcauchy(x)))/m*(2*dcauchy(0)) #2*decauchy(0)     ,sum(as.numeric(y<=dcauc
est
}
mc.hit(m)
```

```
## [1] 0.1461031
```

##control var

```
mccontrol <- function(m){
x <- runif(m, 0,0.5)
g <- dcauchy(x)/2
f1 <- x^2
f2 <- x^4
muf1 <- 1/12
muf2 <- 1/80
L <- lm(g~f1+f2)
#est <- mean(g) - L$coeff[2]*(mean(f1)-muf2) - L$coeff[3]*(mean(f2)-muf2)
est2 <- sum(lm(g ~ f1+f2)$coeff*c(1, muf1, muf2))
est2
}
mccontrol(m)
```

```
## [1] 0.1475834
```

##var 비교

```
set.seed(1)
rep11 <- foreach(i=1:100, .combine = bind_rows) %do%
               {
                 tibble(MC1=mc(m),
                 MC21=mc.impo1(m),
                 MC22=mc.impo2(m),
                 MC3=mc.hit(m),
                 MC4=mccontrol(m))
} %>%
  summarise(base_vs_impo1=sd(MC1)^2/sd(MC21)^2,
            base_vs_impo2=sd(MC1)^2/sd(MC22)^2,
            base_vs_hit=sd(MC1)^2/sd(MC3)^2,
            base_vs_conval=sd(MC1)^2/sd(MC4)^2)



(resault4 <- rep11)
```

```
## # A tibble: 1 x 4
##   base_vs_impo1 base_vs_impo2 base_vs_hit base_vs_conval
##           <dbl>         <dbl>       <dbl>          <dbl>
## 1         1011.         1117.       0.889     122286432.
```