

Script2-16 Monte Carlo Simulation2

Kei Sakamoto

何回も推定し直せばパラメータに対して平均とか分散計算できるので、理論値に近づいてるか確認できるのが script2-14 との違い。

Set the random seed

```
set.seed(1234567) #seed はデキトーにセット
```

set sample size and number of simulations

```
n<-1000; r<-10000 #replications のr。
```

set true parameters: betas and sd of u

```
b0<-1; b1<-0.5; su<-2
```

initialize b0hat and b1hat to store results later

```
b0hat <- numeric(r) #10000 個の 0 で初期化  
b1hat <- numeric(r)
```

Draw a sample of x, fixed over replications(x は毎回同じセット)

```
x <- rnorm(n,4,1)
```

repeat r times

```
for(j in 1:r) {  
  u <- rnorm(n,0,su) #u のセット(と当然y のセット)は毎回変えてその都度パラメータ推定し直し  
  y <- b0 + b1*x + u  
  
  bhat <- coefficients( lm(y~x) )  
  b0hat[j] <- bhat["(Intercept)"] #一個ずつ順番に 0 と置き換えて記録していく。  
  b1hat[j] <- bhat["x"]  
}
```

MC estimate(モンテカルロ推定値) of the expected values

```
mean(b0hat)
```

```
## [1] 0.9985388
```

```
mean(b1hat)
```

```
## [1] 0.5000466
```

どちらも期待値に近づいている。

MC estimate of the variances

```
var(b0hat)
```

```
## [1] 0.0690833
```

```
var(b1hat)
```

```
## [1] 0.004069063
```

それぞれの分散の理論値(0.0685 と 0.004)に近づいている。

Initialize empty plot(毎回違うデータセットを plot するのはカオスなのでいっそなんも plot しないほうが見やすい)

```
plot( NULL, xlim=c(0,8), ylim=c(0,6), xlab="x", ylab="y")
for (j in 1:10) abline(b0hat[j],b1hat[j],col="gray") #最初の10本だけ gray
のfittedを追加
abline(b0,b1,lwd=2) #PRF も追加
legend("topleft",c("PRF","OLS reg fitted"),
      lwd=c(2,1),col=c("black","gray"))
```

