Chapter 5: OLS asymptotics

Susumu Shikano

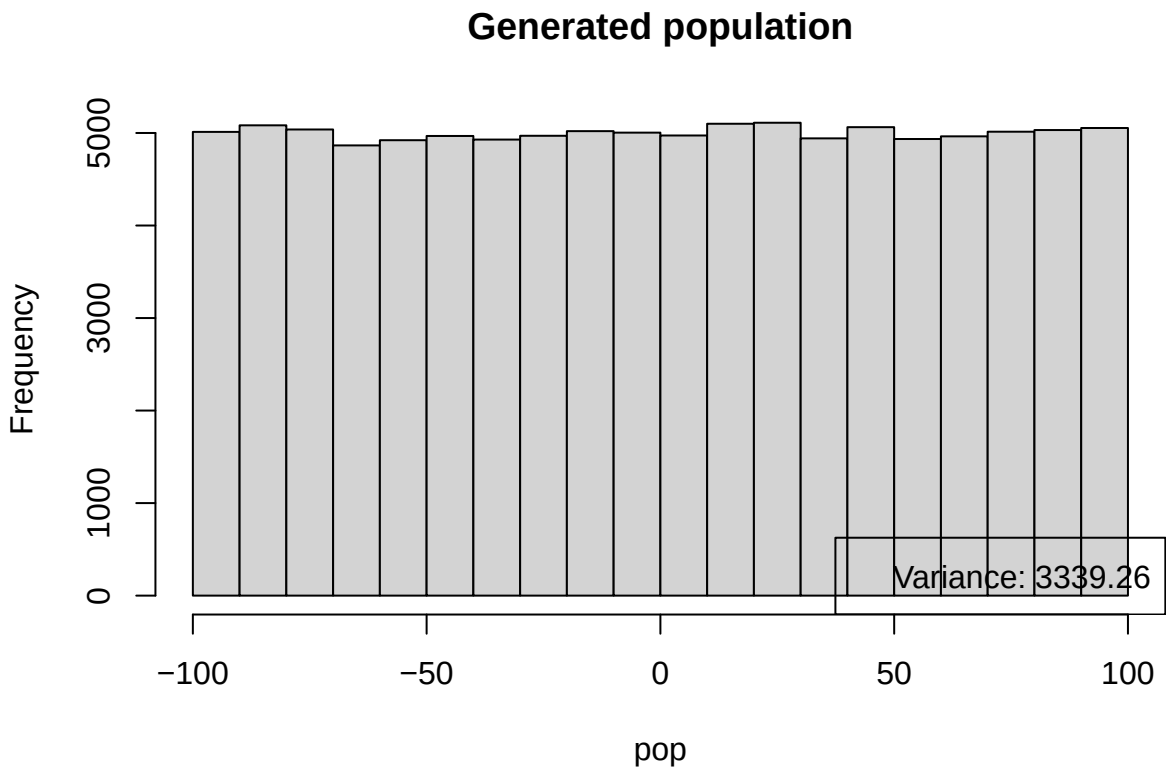
Last compiled at 13. Juli 2022

Sample variance as biased and consistent estimator

Suppose that we are interested to estimate the variance of a population.

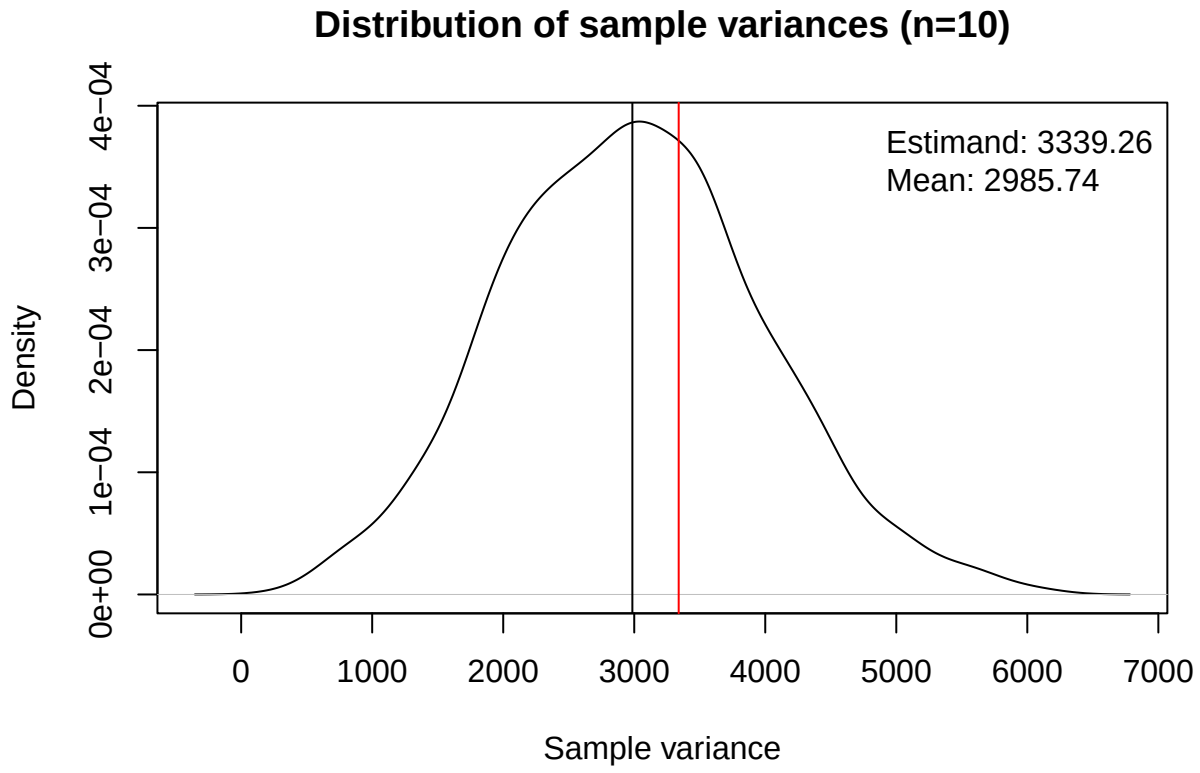
We generate a population by using a uniform distribution and calculate the population variance.

```
pop <- runif(100000,-100,100)

hist(pop,main="Generated population")

pop.var <- naive.var(pop)

legend("bottomright",paste("Variance:", round(pop.var,2)))
```



We draw multiple samples of size n=10 and calculate their sample vaqriances.

```r
n.iter <- 1000
all.sample.var.n10 <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=10)
  this.sample.var <- naive.var(this.sample)
  all.sample.var.n10[i] <- this.sample.var
}

plot(density(all.sample.var.n10),
     main="Distribution of sample variances (n=10)",
     xlab="Sample variance")
abline(v=pop.var,col="red")
abline(v=mean(all.sample.var.n10))
legend("topright",c(paste("Estimand:", round(pop.var,2)),
                    paste("Mean:", round(mean(all.sample.var.n10),2))
                    ),bty="n")
```

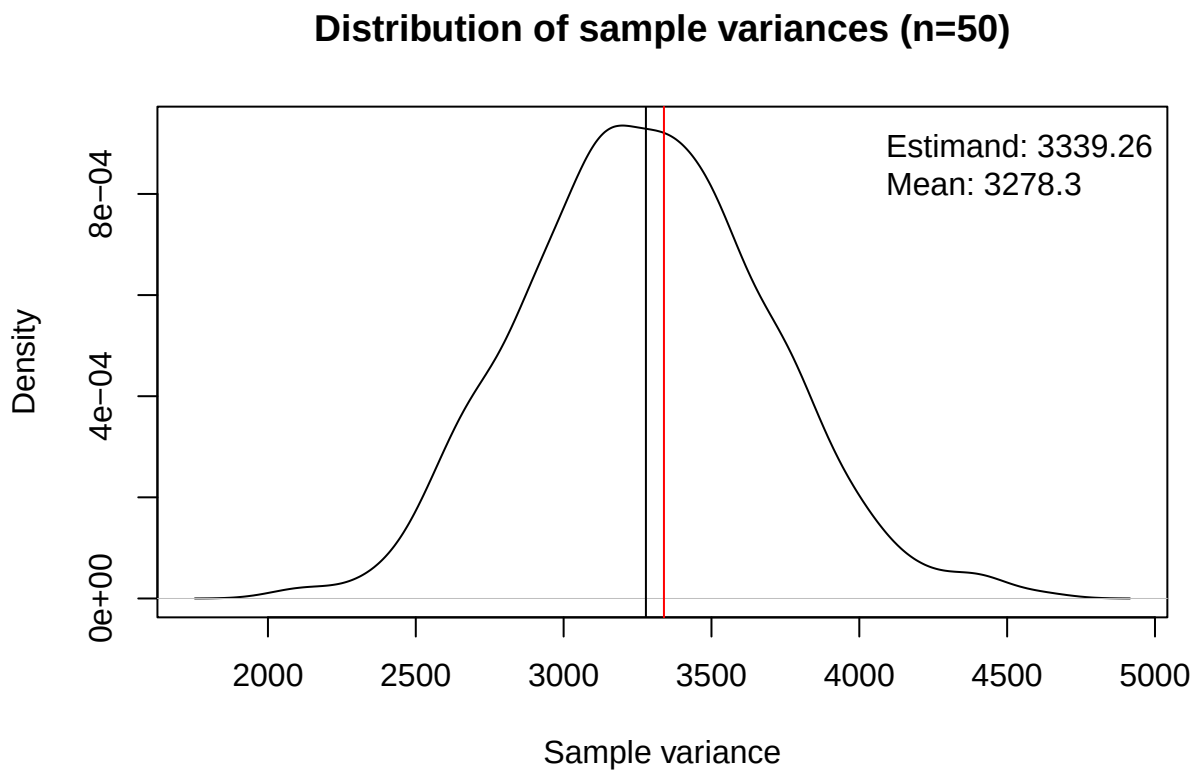


The red vertical line is the population variance to be estimated (3339.26). The black vertical line is the mean sample variance (2985.74). Obviously, the sample variance is biased and tend to underestimate the population variance.

Now we increase the sample size to n=50 and repeat the same exercise.

```r
n.iter <- 1000
all.sample.var.n50 <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=50)
```

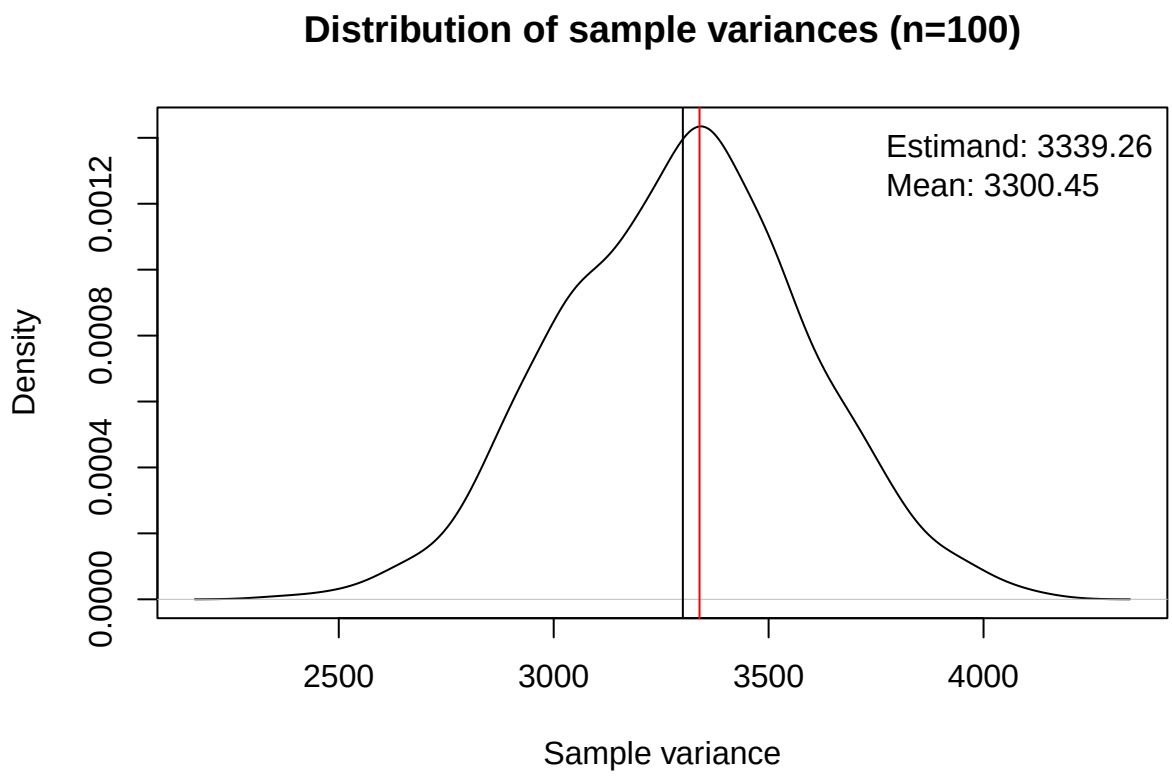
```
  this.sample.var <- naive.var(this.sample)
  all.sample.var.n50[i] <- this.sample.var
}

plot(density(all.sample.var.n50),
     main="Distribution of sample variances (n=50)",
     xlab="Sample variance")
abline(v=pop.var,col="red")
abline(v=mean(all.sample.var.n50))
legend("topright",c(paste("Estimand:", round(pop.var,2)),
                    paste("Mean:", round(mean(all.sample.var.n50),2))
                    ),bty="n")
```



We draw multiple samples of size n=100 and calculate their sample mean.

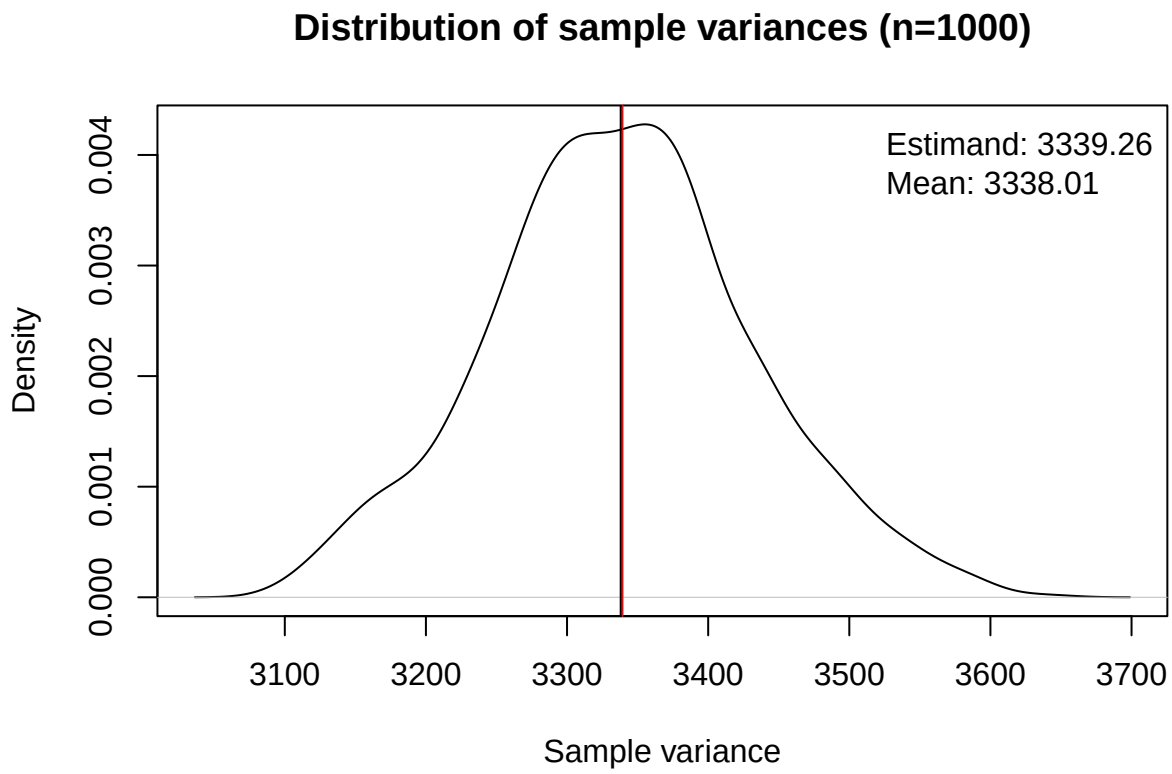
```
n.iter <- 1000
all.sample.var.n100 <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=100)
  this.sample.var <- naive.var(this.sample)
  all.sample.var.n100[i] <- this.sample.var
}

plot(density(all.sample.var.n100),
     main="Distribution of sample variances (n=100)",
     xlab="Sample variance")
```

```
abline(v=pop.var,col="red")
abline(v=mean(all.sample.var.n100))
legend("topright",c(paste("Estimand:", round(pop.var,2)),
                    paste("Mean:", round(mean(all.sample.var.n100),2))
                    ),bty="n")
```



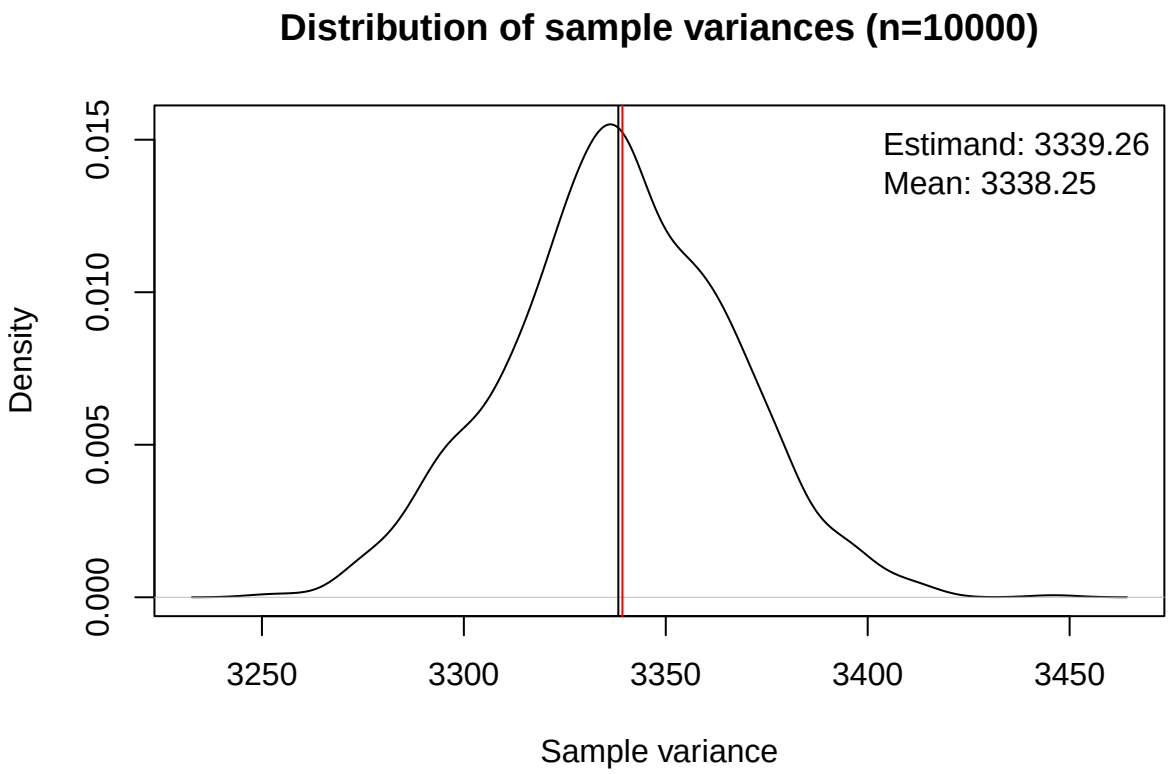
We draw multiple samples of size n=1000 and calculate their sample mean.

```
n.iter <- 1000
all.sample.var.n1000 <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=1000)
  this.sample.var <- naive.var(this.sample)
  all.sample.var.n1000[i] <- this.sample.var
}

plot(density(all.sample.var.n1000),
     main="Distribution of sample variances (n=1000)",
     xlab="Sample variance")
abline(v=pop.var,col="red")
abline(v=mean(all.sample.var.n1000))
legend("topright",c(paste("Estimand:", round(pop.var,2)),
                    paste("Mean:", round(mean(all.sample.var.n1000),2))
                    ),bty="n")
```



We draw multiple samples of size n=10000 and calculate their sample mean.

```
n.iter <- 1000
all.sample.var.n10000 <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=10000)
  this.sample.var <- naive.var(this.sample)
  all.sample.var.n10000[i] <- this.sample.var
}

plot(density(all.sample.var.n10000),
     main="Distribution of sample variances (n=10000)",
     xlab="Sample variance")
abline(v=pop.var,col="red")
abline(v=mean(all.sample.var.n10000))
legend("topright",c(paste("Estimand:", round(pop.var,2)),
                    paste("Mean:", round(mean(all.sample.var.n10000),2))
                    ),bty="n")
```

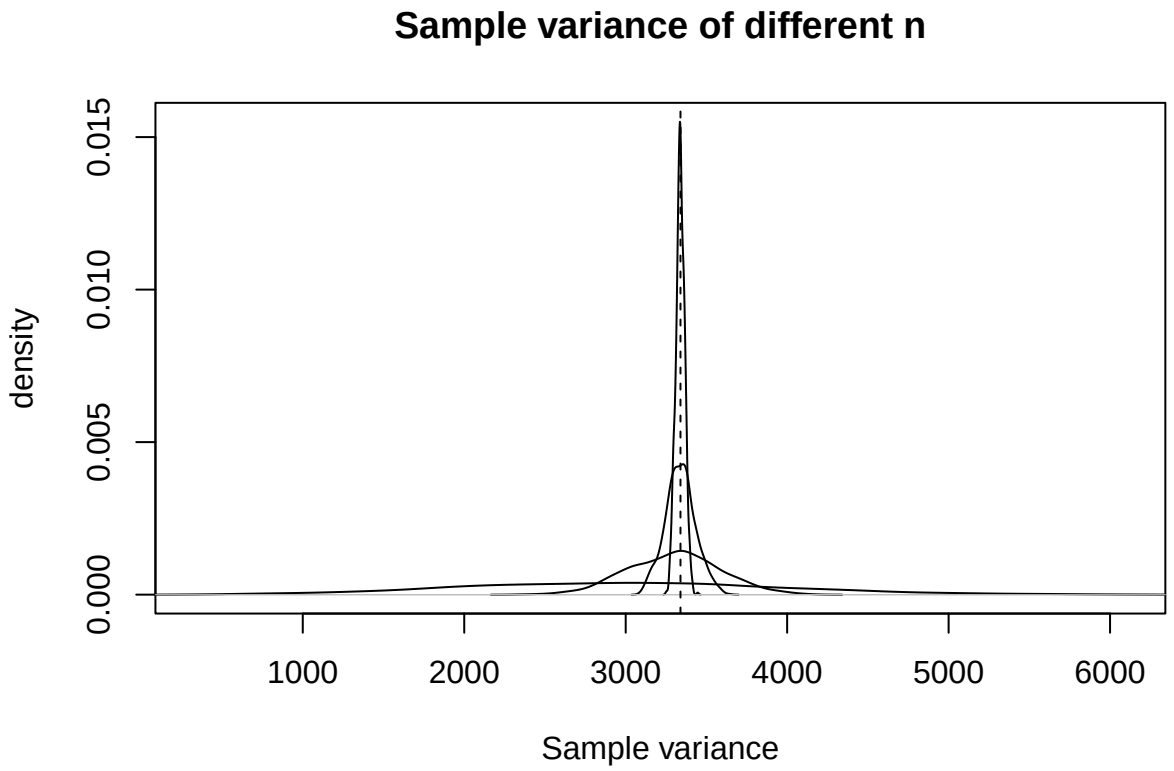


Plot all distributions

There is no clear differences between the distributions above. It becomes obvious if we plot all distribution in the same figure:

```
range.x <- range(c(all.sample.var.n10000,
                   all.sample.var.n1000,
                   all.sample.var.n100,
                   all.sample.var.n10))

max.y <- max(c(density(all.sample.var.n10000)$y,
                 density(all.sample.var.n1000)$y,
                 density(all.sample.var.n100)$y,
                 density(all.sample.var.n10)$y))

plot(0,type="n",xlim=range.x,ylim=c(0,max.y),
     main="Sample variance of different n",
     xlab="Sample variance",
     ylab="density")
par(new=T)
plot(density(all.sample.var.n10000),xlim=range.x,ylim=c(0,max.y),ann=F,axes=F)
par(new=T)
plot(density(all.sample.var.n1000),xlim=range.x,ylim=c(0,max.y),ann=F,axes=F)
par(new=T)
plot(density(all.sample.var.n100),xlim=range.x,ylim=c(0,max.y),ann=F,axes=F)
par(new=T)
```

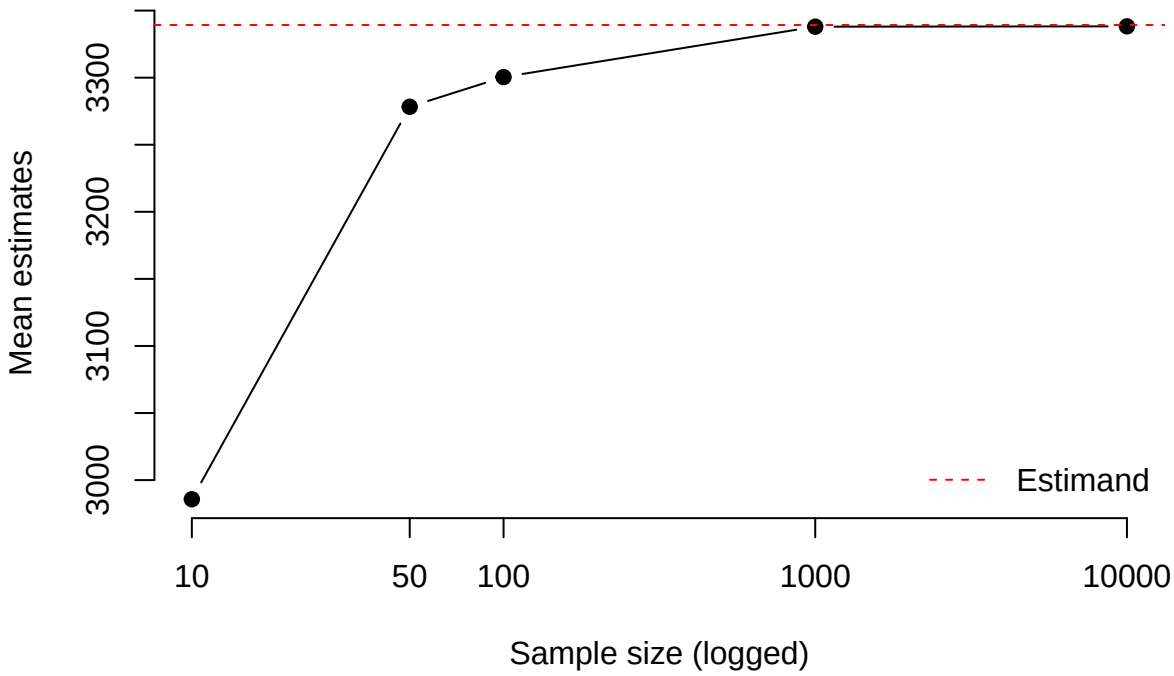
```
plot(density(all.sample.var.n10),xlim=range.x,ylim=c(0,max.y),ann=F,axes=F)
abline(v=pop.var,lty=2)
```



It is clearly to see that sample variances come closer to the population variance as n increases. That is, the sample variance is a consistent estimator while it is biased.

The bias is visualized in the below figure:

```
sample.var.means <- c(
      mean(all.sample.var.n10),
      mean(all.sample.var.n50),
      mean(all.sample.var.n100),
      mean(all.sample.var.n1000),
      mean(all.sample.var.n10000))

plot(log(c(10,50,100,1000,10000)),sample.var.means,type="b",pch=19,
     axes=F,
     xlab="Sample size (logged)",
     ylab="Mean estimates")
axis(2)
axis(1,at=log(c(10,50,100,1000,10000)),c(10,50,100,1000,10000))
abline(h=pop.var,lty=2,col="red")
legend("bottomright",lty=2,col="red",c("Estimand"),bty="n")
```

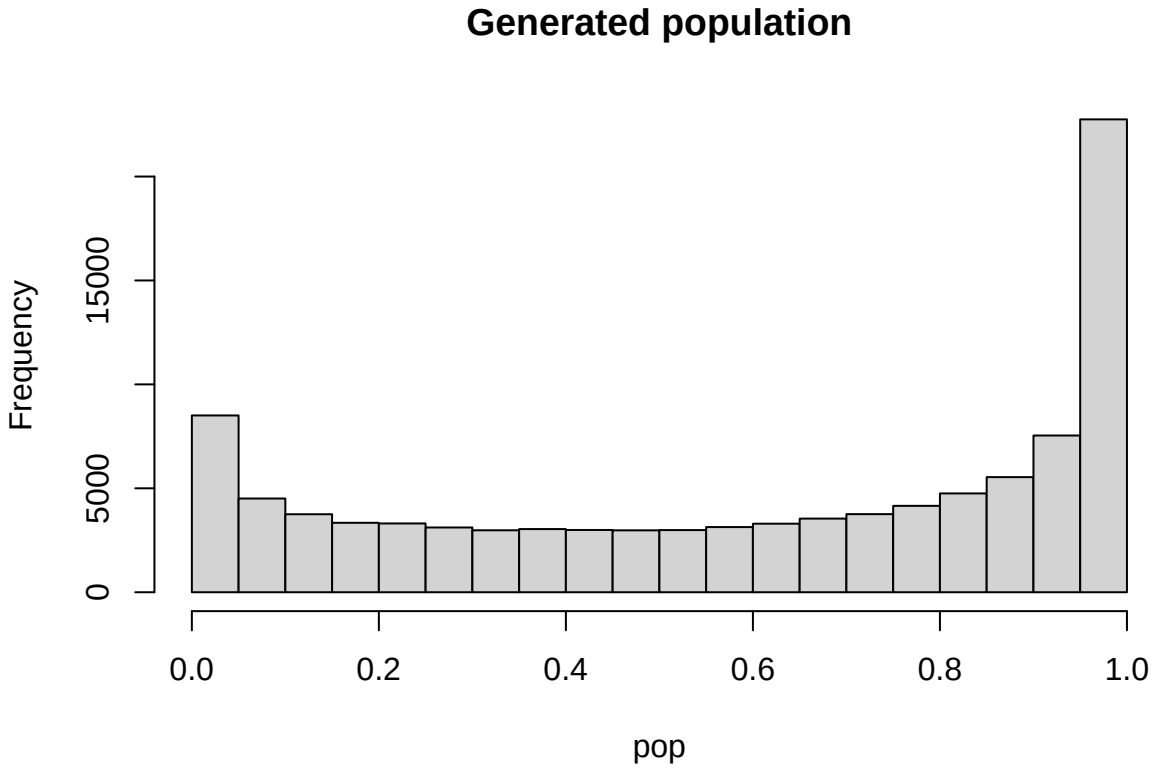


This figure demonstrates that the amount of bias decreases in increasing sample sizes. The bias can be corrected by the factor n/(n-1), which converges to the limit 1 as n increases. In other words, the bias is ignorable for a large sample size. Together with the decreasing standard error, all estimates converge to the estimand (consistency). However, even though we can ignore the bias for a large sample size, the estimator is still biased.

Central limit theorem

First generate a population with two extreme values.

```
pop <- rbeta(100000,0.6,0.4)
hist(pop,main="Generated population")
```



You can check the mean and variance of this population:

```
pop.mean <- mean(pop)
pop.var <- naive.var(pop)

pop.mean
```

```
## [1] 0.5990659
```

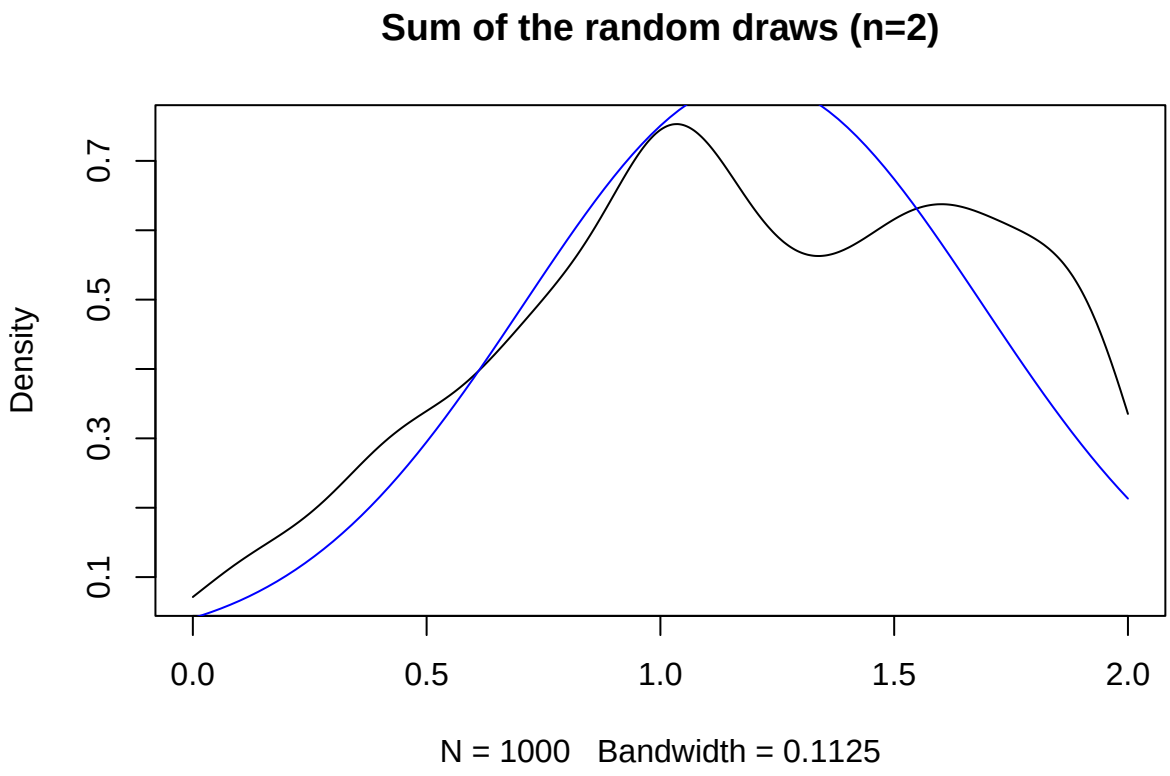
```
pop.var
```

```
## [1] 0.1198834
```

From this population, we can draw multiple samples with size of 2, calculate the sample sum and observe its distribution.

```
n.iter <- 1000
sample.size <- 2
all.sample.sum <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=sample.size)
  this.sample.sum <- sum(this.sample)
  all.sample.sum[i] <- this.sample.sum
}

plot(density(all.sample.sum,from=0,to=sample.size),
     main=paste0("Sum of the random draws (n=",sample.size,")"))
par(new=T)
this.norm <- function(x) dnorm(x,
```

```
                                  mean=pop.mean*sample.size,
                                  sd=sqrt(sample.size*pop.var))
curve(this.norm,0,sample.size,add=T,col="blue")
```




```
mean(all.sample.sum)
```

```
## [1] 1.185515
```

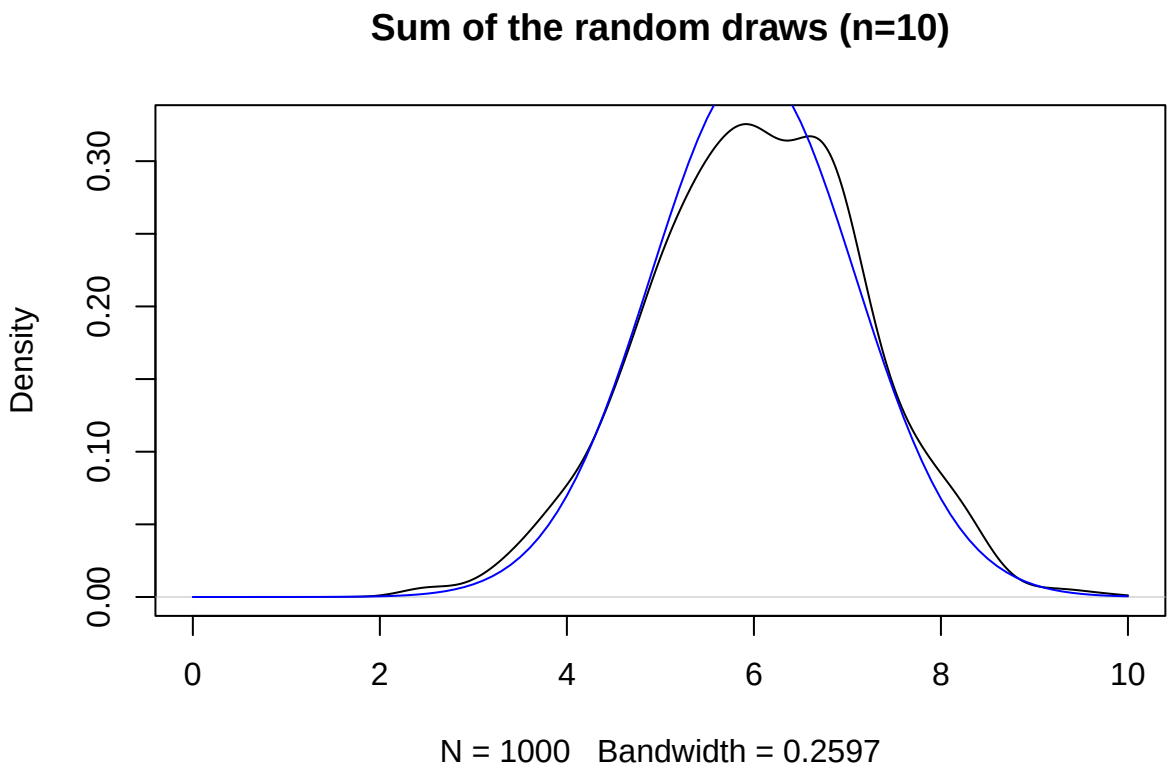
```
var(all.sample.sum)
```

```
## [1] 0.2474585
```

From this population, we can draw multiple samples with size of 10, calculate the sample sum and observe its distribution.

```
n.iter <- 1000
sample.size <- 10
all.sample.sum <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=sample.size)
  this.sample.sum <- sum(this.sample)
  all.sample.sum[i] <- this.sample.sum
}

plot(density(all.sample.sum,from=0,to=sample.size),
     main=paste0("Sum of the random draws (n=",sample.size,")"))
par(new=T)
this.norm <- function(x) dnorm(x,
```

```
                                  mean=pop.mean*sample.size,
                                  sd=sqrt(sample.size*pop.var))
curve(this.norm,0,sample.size,add=T,col="blue")
```




```
mean(all.sample.sum)
```

```
## [1] 6.01918
```

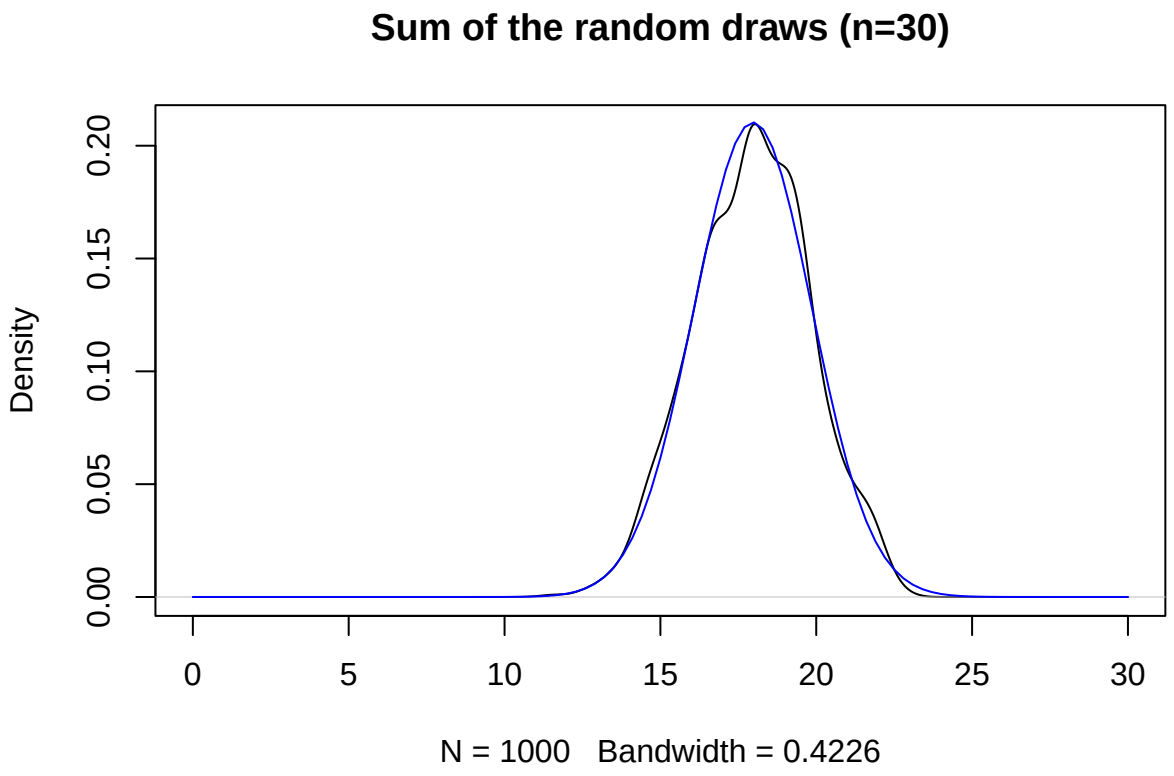
```
var(all.sample.sum)
```

```
## [1] 1.319285
```

From this population, we can draw multiple samples with size of 30, calculate the sample sum and observe its distribution.

```
n.iter <- 1000
sample.size <- 30
all.sample.sum <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=sample.size)
  this.sample.sum <- sum(this.sample)
  all.sample.sum[i] <- this.sample.sum
}

plot(density(all.sample.sum,from=0,to=sample.size),
     main=paste0("Sum of the random draws (n=",sample.size,")"))
par(new=T)
this.norm <- function(x) dnorm(x,
```

```
                                  mean=pop.mean*sample.size,
                                  sd=sqrt(sample.size*pop.var))
curve(this.norm,0,sample.size,add=T,col="blue")
```




```
mean(all.sample.sum)
```

```
## [1] 17.95277
```

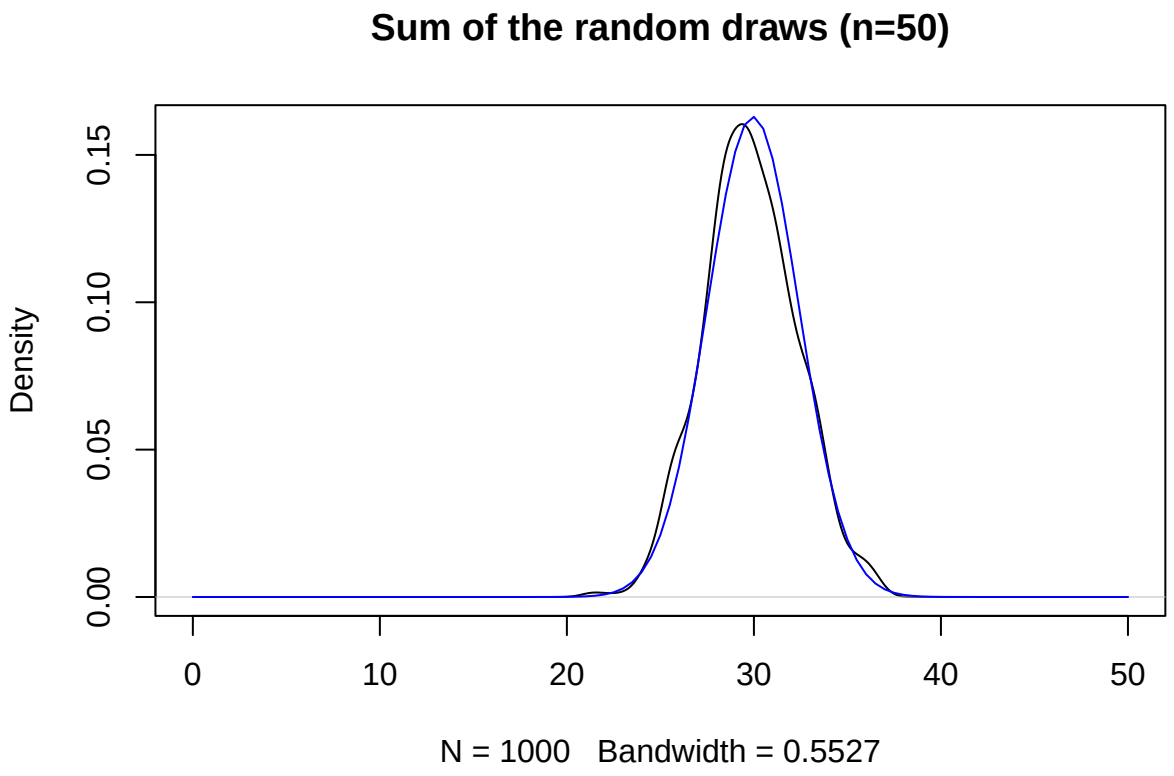
```
var(all.sample.sum)
```

```
## [1] 3.494178
```

From the population, we can draw multiple samples with size of 50, calculate the sample sum and observe its distribution.

```
n.iter <- 1000
sample.size <- 50
all.sample.sum <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=sample.size)
  this.sample.sum <- sum(this.sample)
  all.sample.sum[i] <- this.sample.sum
}

plot(density(all.sample.sum,from=0,to=sample.size),
     main=paste0("Sum of the random draws (n=",sample.size,")"))
par(new=T)
this.norm <- function(x) dnorm(x,
```

```
                                  mean=pop.mean*sample.size,
                                  sd=sqrt(sample.size*pop.var))
curve(this.norm,0,sample.size,add=T,col="blue")
```




```
mean(all.sample.sum)
```

```
## [1] 29.81334
```

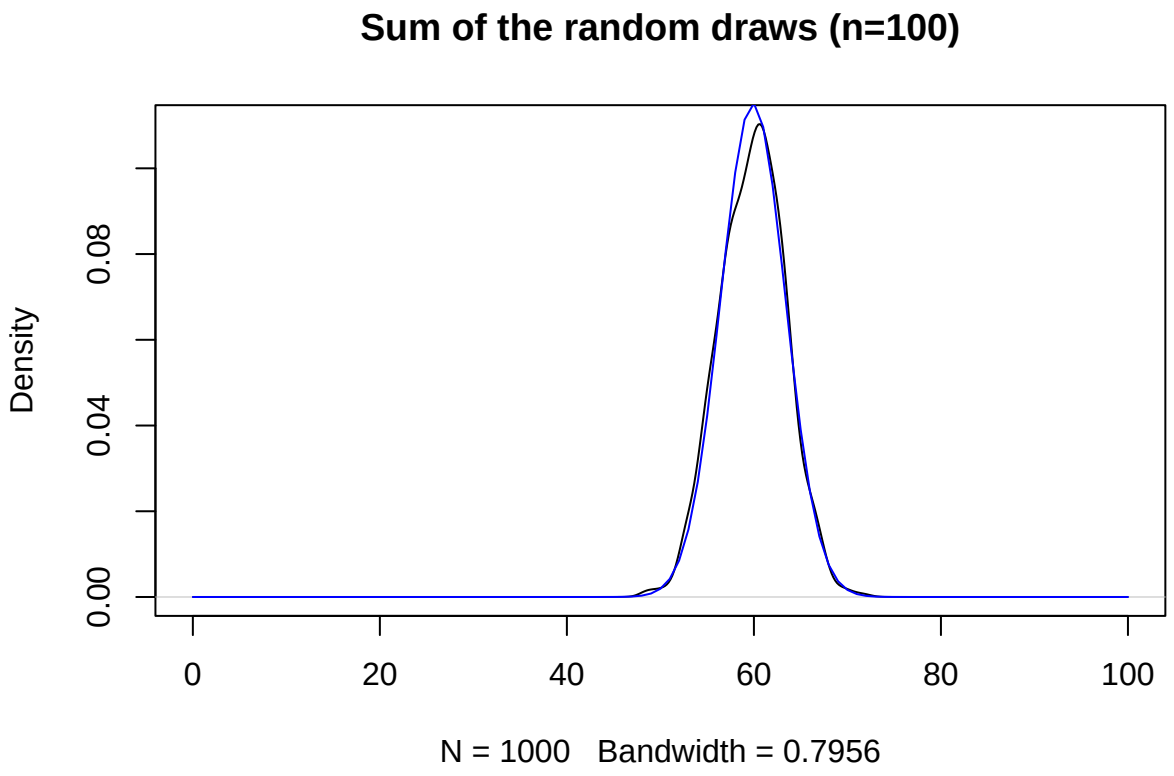
```
var(all.sample.sum)
```

```
## [1] 6.204545
```

From the population, we can draw multiple samples with size of 100, calculate the sample sum and observe its distribution.

```
n.iter <- 1000
sample.size <- 100
all.sample.sum <- rep(NA,n.iter)
for (i in 1:n.iter){
  this.sample <- sample(pop,size=sample.size)
  this.sample.sum <- sum(this.sample)
  all.sample.sum[i] <- this.sample.sum
}

plot(density(all.sample.sum,from=0,to=sample.size),
     main=paste0("Sum of the random draws (n=",sample.size,")"))
par(new=T)
this.norm <- function(x) dnorm(x,
```

```
                                  mean=pop.mean*sample.size,
                                  sd=sqrt(sample.size*pop.var))
curve(this.norm,0,sample.size,add=T,col="blue")
```




```
mean(all.sample.sum)
```

```
## [1] 59.84143
```

```
var(all.sample.sum)
```

```
## [1] 12.38504
```

Above, with increasing number of observations, the distribution of the sample sum becomes closer to a normal distribution. And the mean of this sample sum is identical with the population mean, if we draw an infinitely large number of samples. The variance is identical with the population variance times the sample size.

Consistency of the OLS estimator

We generate 1000 datasets with different sample sizes (n= 5, n= 10, n= 25, n= 100 under the GM-assumptions. The number of independent variables is 2. The true regression line has the intercept of 1 and the slope of 5, -2.5. The independent variables are generated with the mean 2, -1, variances 3, 5 and covariance -1. We further assume uniformly distributed errors with the variance 100.

```
unif.range <- sqrt(true.err.var*12) # transform the variance into the range

all.generated.samples <- vector(mode="list",length = length(sample.sizes))
for (i in 1:length(sample.sizes)){
  all.generated.samples[[i]] <- data.generation(sample.size=sample.sizes[i],
```

```
                                n.sim=num.datasets,
                                n.iv=n.iv,
                                x.mu=x.mu,
                                x.Sigma=x.Sigma,
                                para=c(true.intercept,true.slope),
                                err.dist = "uniform",
                                err.disp = unif.range)
}
```

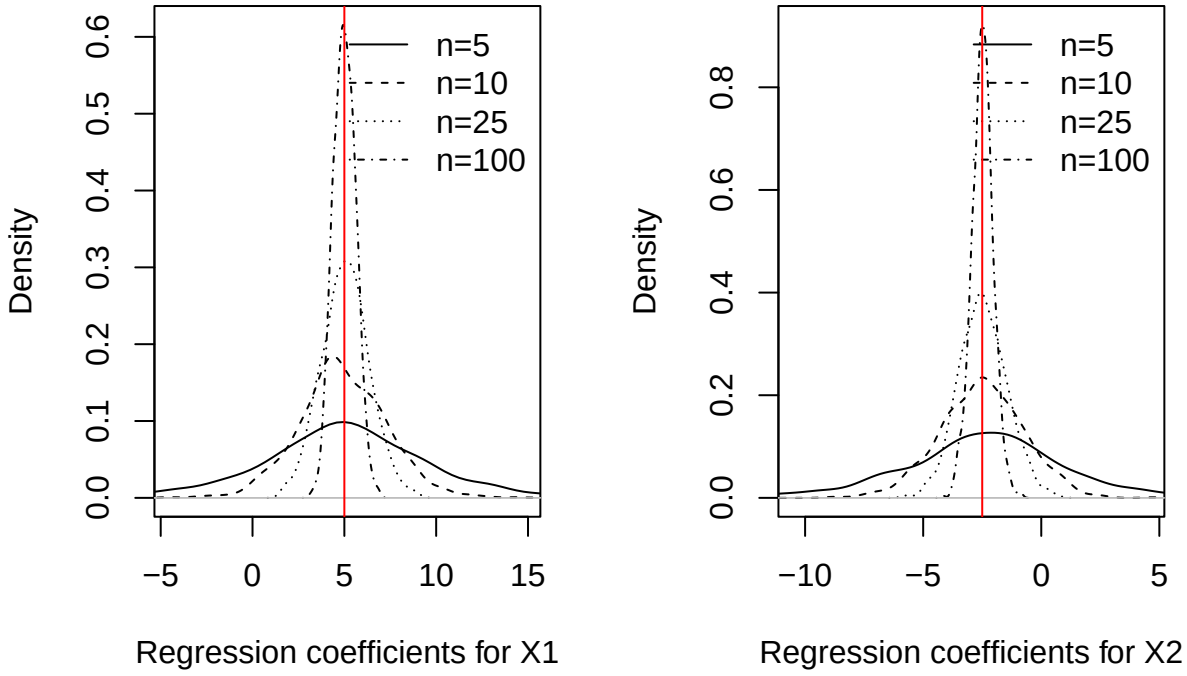
We repeat the multiple regression analysis by using each of 1000 datasets:

```
all.coef <- array(NA,dim=c(num.datasets,4,length(sample.sizes)))
all.coef.se <- array(NA,dim=c(num.datasets,3,length(sample.sizes)))

for (i.data in 1:length(sample.sizes)){
  this.samples <- all.generated.samples[[i.data]]
for (i in 1:num.datasets){

    this.data <- this.samples$generated.data[[i]]

    lm.out <- lm(y ~ X1 + X2 ,data=  this.data)

    all.coef[i,1:3,i.data] <- coef(lm.out)
    all.coef[i,4,i.data] <- summary(lm.out)$sigma
    all.coef.se[i,,i.data] <- coef(summary(lm.out))[,2]


}
}

dimnames(all.coef)[[2]] <- c("b0","b1","b2","sigma")
dimnames(all.coef.se)[[2]] <- c("b0","b1","b2")
```

We can now plot all the estimates of each coefficients in the same figure.

```
par(mfrow=c(1,2))

for (i.fig in 1:2){
    #x.range <- range(c(all.coef[,c("b1","b2")[i.fig],]))
    x.range <-quantile(c(all.coef[,c("b1","b2")[i.fig],]),pr=c(0.01,0.99))

    plot(density.out <- density(all.coef[,c("b1","b2")[i.fig],length(sample.sizes)]),
         main="",xlab=paste0("Regression coefficients for X",i.fig),
         xlim=x.range,lty=length(sample.sizes))
    #abline(v=mean(all.coef[,c("b1","b2")[i.fig],1]))

    for (i.data in (length(sample.sizes)-1):1){
    par(new=T)
    plot(density(all.coef[,c("b1","b2")[i.fig],i.data]),ann=F,axes=F,
         xlim=x.range,lty=i.data,
         ylim=c(0,max(density.out$y)))
    #abline(v=mean(all.coef[,c("b1","b2")[i.fig],i.data]),lty=2)

    }
    legend("topright",lty=c(1:length(sample.sizes)),
```

```
           paste0("n=",sample.sizes),bty="n")
    abline(v=true.slope[i.fig],col="red")
}
```



The figure demonstrates that the estimates converges to the true parameter values (the red vertical lines) as n increases.

Asymptotic normality and large sample inference

We have generated the datasets by assuming the uniformly distributed errors. Therefore, the inferece based on t- or F-distribution is not exact. However, increasing n allow us to use them approximately.

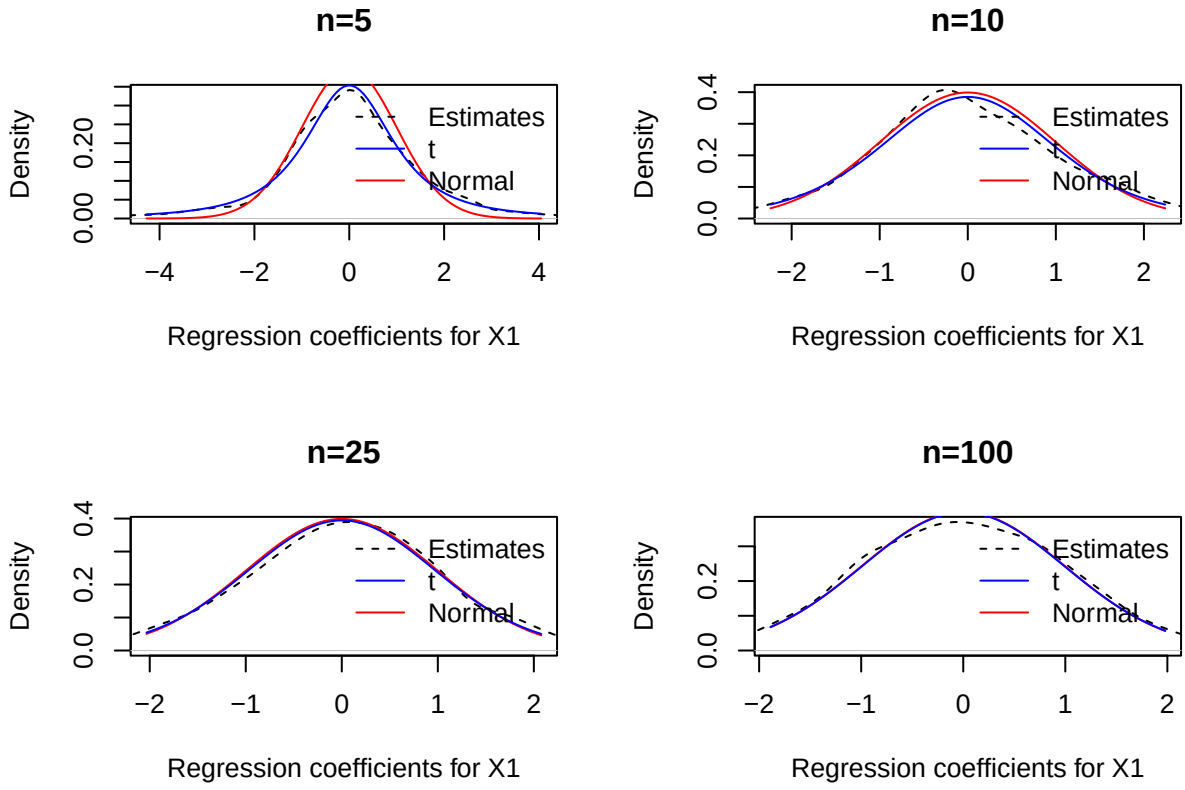
To see this, we can first calculate the t-value for all regression results above:

```
all.coef.norm <- all.coef[,1:3,]
all.coef.norm[,1,] <- all.coef.norm[,1,] - true.intercept
all.coef.norm[,2,] <- all.coef.norm[,2,] - true.slope[1]
all.coef.norm[,3,] <- all.coef.norm[,3,] - true.slope[2]

all.coef.norm <- all.coef.norm/all.coef.se

dimnames(all.coef.norm)[[2]] <- c("b0","b1","b2")
```

... and observe their distributions:

```
par(mfrow=c(2,2))

for (i.fig in 1:length(sample.sizes)){
```

```
    x.range <-quantile(c(all.coef.norm[,c("b1","b2")[1],i.fig]),pr=c(0.025,0.975))

    plot(density(all.coef.norm[,c("b1","b2")[1],i.fig]),
         main=paste0("n=",sample.sizes[i.fig]),
         xlab=paste0("Regression coefficients for X",1),
         xlim=x.range,
         lty=2)
    #abline(v=mean(all.coef[,c("b1","b2")[i.fig],1]))

    #legend("topright",lty=c(1:length(sample.sizes)),
    #       paste0("n=",sample.sizes),bty="n")
    #abline(v=true.slope[1],col="red")

    curve(dnorm,add=TRUE,col="red")
    this.dt <- function(x) dt(x,df=sample.sizes[i.fig]-3)
    curve(this.dt,add=TRUE,col="blue")

    legend("topright",
           col=c("black","blue","red"),
           lty=c(2,1,1),
           legend=c("Estimates","t","Normal"),
           bty="n")
}
```



For a small sample size, the distribution of the empirical estimates strongly deviates from the standard normal distribution. As n increases, however, all the distributions come closer to each other. If n approaches

infinity, the distribution of the OLS estimates converges to the standard normal distribution, which is called asymptotic normality of the OLS estimator.

Asymptotic Efficiency of the OLS estimator

Suppose that we are interested to estimate a simple regression model by using two different estimators: OLS and another alternative estimator.

The alternative estimator to be compared here is

$$\tilde{\beta}_1 = \frac{\sum(z_i - \bar{z})y_i}{\sum(z_i - \bar{z})x_i}$$

with

$$z_i = \frac{1}{1 + |x_i|}$$

```
g.func <- function(x) 1/(1+ abs(x))

alternative.estimator <- function(y,x){
  slope <- sum((g.func(x)-mean(g.func(x)))*y )/sum((g.func(x)-mean(g.func(x)))*x )
  intercept <- mean(y) - slope * mean(x)
  c(intercept,slope)
}
```

This estimator is consistent under the GM-assumption (see Wooldridge).

To check the performance of both estimators, we generate 2000 datasets with different sample sizes (n= 5, n= 100, n= 1000 under the GM-assumptions. The number of independent variables is 1. The true regression line has the intercept of 1 and the slope of 5. The independent variables are generated with the mean 2, variances 3. We further assume uniformly distributed errors with the variance 100.

```
unif.range <- sqrt(true.err.var*12) # transform the variance into the range

all.generated.samples <- vector(mode="list",length = length(sample.sizes))
for (i in 1:length(sample.sizes)){
  all.generated.samples[[i]] <- data.generation(sample.size=sample.sizes[i],
                                n.sim=num.datasets,
                                n.iv=n.iv,
                                x.mu=x.mu,
                                x.Sigma=x.Sigma,
                                para=c(true.intercept,true.slope),
                                err.dist = "uniform",
                                err.disp = unif.range)
}
```

We can obtain the OLS estimates by using the first generated sample:

```
data.1 <- all.generated.samples[[2]]$generated.data[[1]]
lm.out <- lm(y ~ X1,data=data.1)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ X1, data = data.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -18.4608  -9.1662   0.1364   9.8926  16.2622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3163     1.5159   1.528     0.13
## X1            4.7621     0.6077   7.836 5.66e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.38 on 98 degrees of freedom
## Multiple R-squared:  0.3852, Adjusted R-squared:  0.3789
## F-statistic: 61.41 on 1 and 98 DF,  p-value: 5.662e-12
```

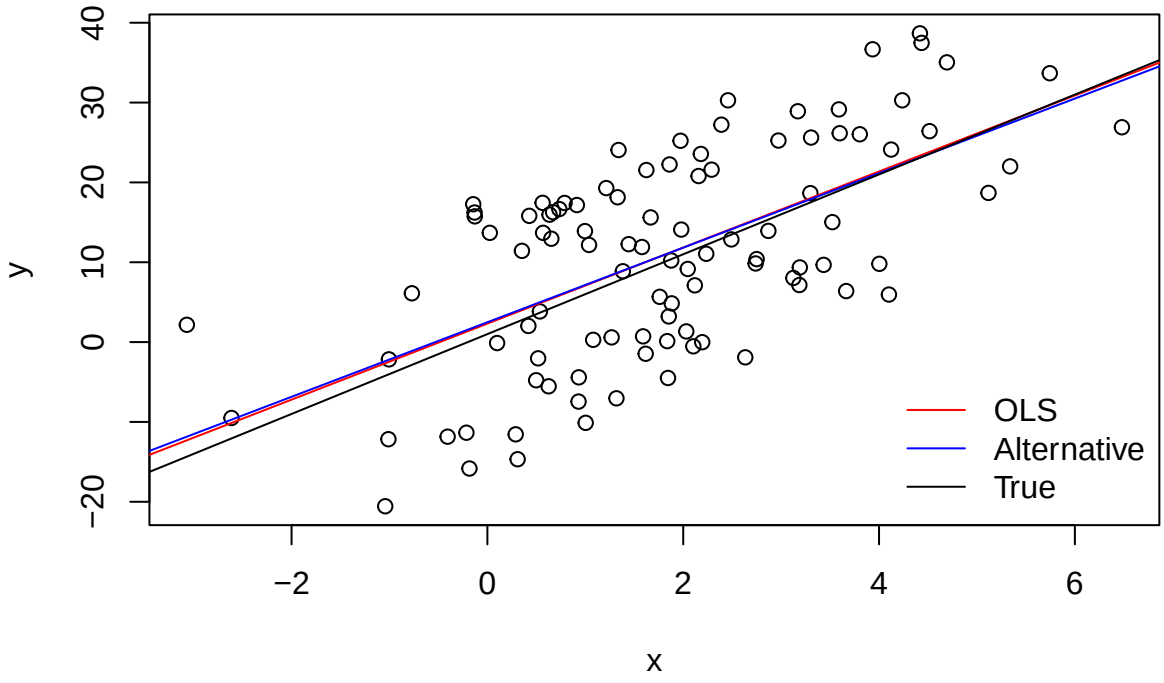
... and the estimates based on the alternative estimator:

```
print(alternative.out <- alternative.estimator(data.1$y , data.1$X1))
```

```
## [1] 2.484814 4.669368
```

We plot the estimated regression lines and the true line in the joint distribution of y and x.

```
plot(data.1$y ~ data.1$X1,ylab="y",xlab="x")
abline(reg=lm.out,col="red")

abline(coef=alternative.out,col="blue")


abline(coef=all.generated.samples[[1]]$para)


legend("bottomright",
       lty=1,
       col=c("red","blue","black"),
       c("OLS","Alternative","True"),
       bty="n")
```

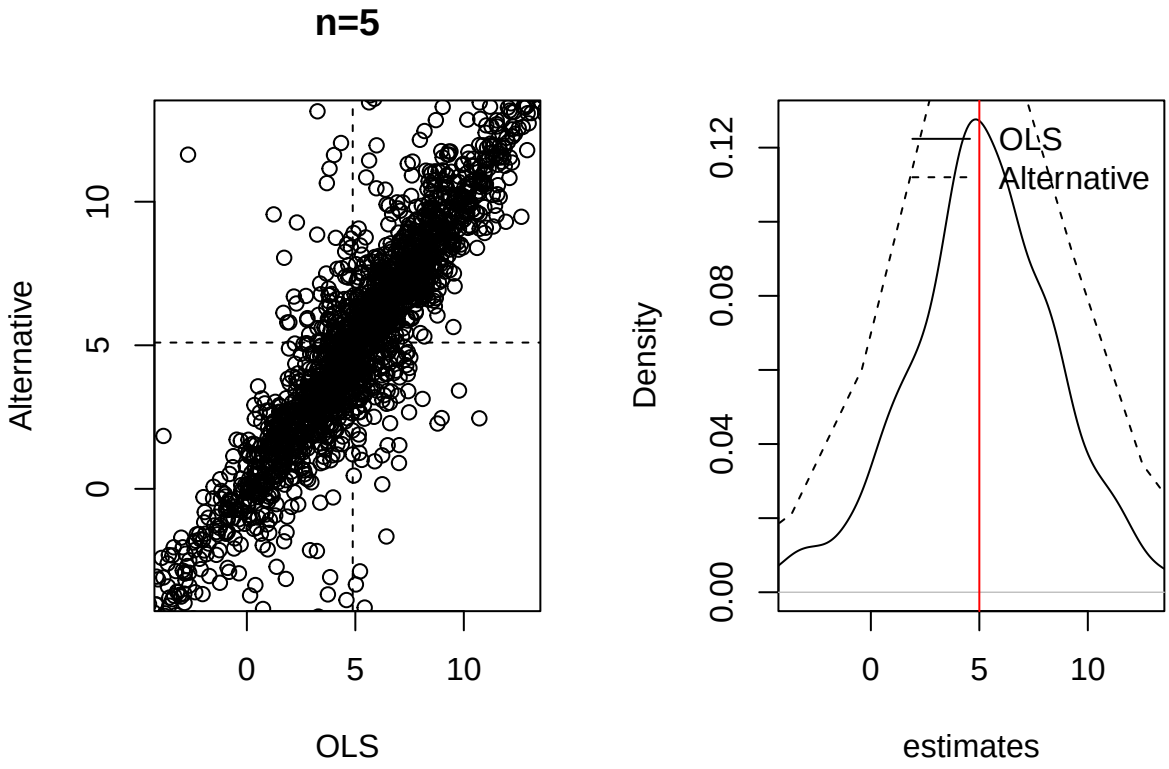


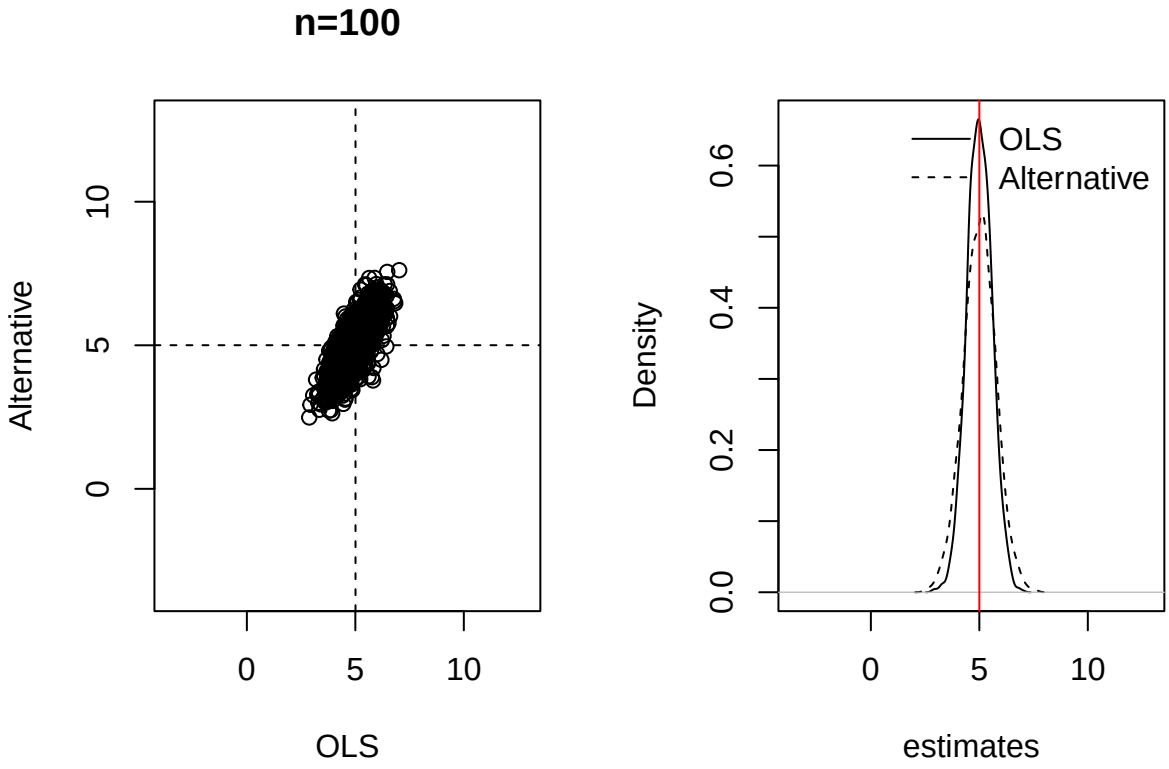
We repeat the regression analysis by using each of 2000 datasets:

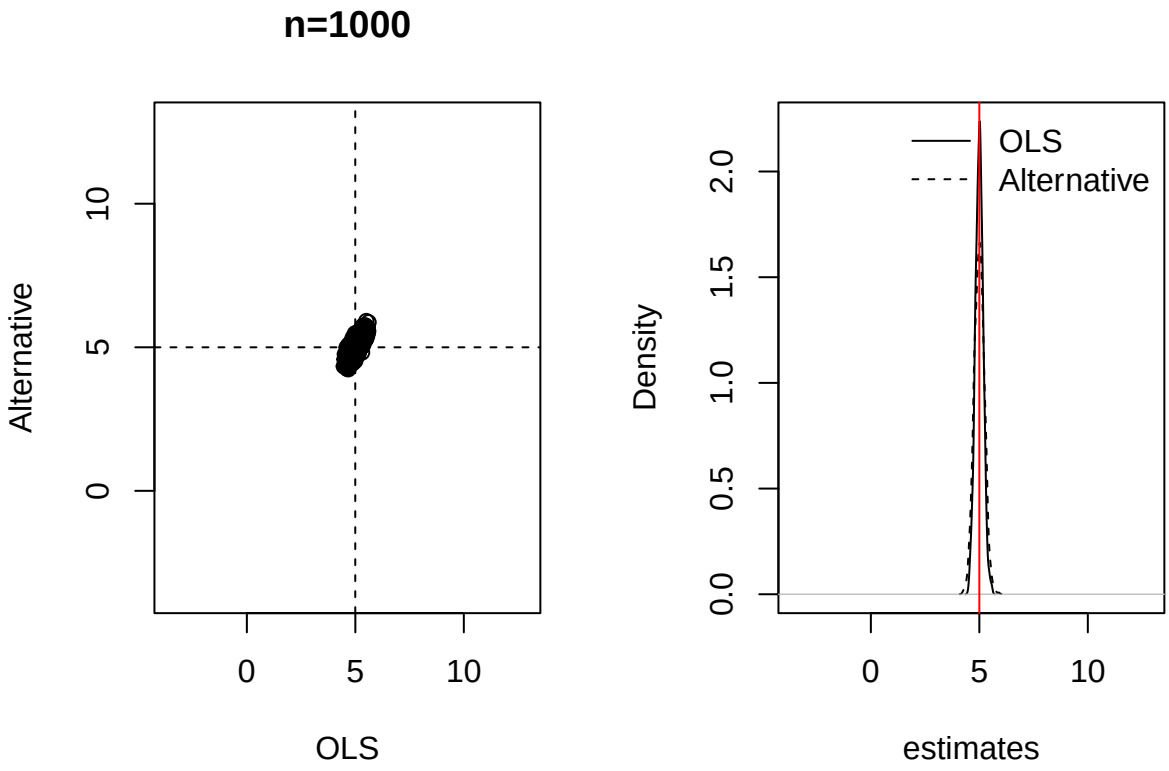
```
all.coef <- array(NA,dim=c(num.datasets,3,length(sample.sizes),2))
all.coef.se <- array(NA,dim=c(num.datasets,2,length(sample.sizes),2))

for (i.data in 1:length(sample.sizes)){
  this.samples <- all.generated.samples[[i.data]]
for (i in 1:num.datasets){

    this.data <- this.samples$generated.data[[i]]

    lm.out <- lm(y ~ X1  ,data=  this.data)

    all.coef[i,1:2,i.data,1] <- coef(lm.out)
    all.coef[i,3,i.data,1] <- summary(lm.out)$sigma
    all.coef.se[i,,i.data,1] <- coef(summary(lm.out))[,2]

    all.coef[i,1:2,i.data,2] <- alternative.estimator(this.data$y,this.data$X1)

}
}

dimnames(all.coef)[[2]] <- c("b0","b1","sigma")
dimnames(all.coef.se)[[2]] <- c("b0","b1")
```

We can observe the joint distribution of the estimates based on both estimators (the left-hand panels) and their marginal distributions (the right-hand panels):

```
#par(mfrow=c(3,2))

for (i.fig in c(1:3)){

  par(mfrow=c(1,2))

  ranges <- quantile(all.coef[,2,,],pr=c(0.01,0.99))

  plot(all.coef[,2,i.fig,2] ~ all.coef[,2,i.fig,1],
     xlab="OLS",ylab="Alternative",
     xlim=ranges,ylim=ranges,
     main=paste0("n=",sample.sizes[i.fig]))
  abline(h=mean(all.coef[,2,i.fig,1]),
       v=mean(all.coef[,2,i.fig,2]),
       lty=2)

#if (i.fig==1)
  x.range <- quantile(all.coef[,2,,],pr=c(0.01,0.99))
plot(density.out <- density(all.coef[,2,i.fig,1]),xlim=x.range,
     xlab="estimates",
     main="")
par(new=T)
plot(density(all.coef[,2,i.fig,2]),
     xlim=x.range,ylim=c(0,max(density.out$y)),
     ann=F,axes=F,
     lty=2)
abline(v=true.slope,col="red")

legend("topright",lty=c(1,2),c("OLS","Alternative"),bty="n")
}
```


n=5
Alternative
OLS
0
5
10
Density
estimates
0.00
0.04
0.08
0.12
OLS
Alternative


n=100
Alternative
OLS
0
5
10
Density
estimates
OLS
Alternative
0.0
0.2
0.4
0.6



From the joint distributions, we can see that both estimators are unbiased. If we look at the marginal distributions, we first see that both estimators are consistent since both distributions converge to the true slope (the red vertical line). At the same time, we can also see that the OLS estimates have always smaller variance than the alternative estimates, which demonstrates the efficiency of the OLS estimator.