Chapter 3: Multiple Regression Analysis: Estimation

Susumu Shikano

Last compiled at 13. Juli 2022

Simple Regression under the GM-assumptions

We generate 5000 datasets with n=500 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 variance of the error is 100. The independent variables are generated with the mean 2, -1, variances 3, 5 and covariance -1.

```
GM.samples <- data.generation(sample.size=sample.size,
                              n.sim=num.datasets,
                              n.iv=n.iv,
                              x.mu=x.mu,
                              x.Sigma=x.Sigma,
                              para=c(true.intercept,true.slope),
                              err.dist = "normal",
                              err.disp = true.err.var)
```

Describing a generated dataset

The first generated dataset looks as follows:

```
data.1 <- GM.samples$generated.data[[1]]

head(data.1)
```

```
##           y         X1         X2      error
## 1  3.781953  0.9267270  0.9131627  0.4312247
## 2 33.091159  4.0682221 -0.4250721 10.6873683
## 3 -8.418451 -2.4644423 -1.2142506 -0.1318662
## 4 19.790591  2.5916852  2.2815470 11.5360324
## 5 32.222867  2.7968318 -0.6653861 15.5752431
## 6 10.760861 -0.3293429 -5.1183718 -1.3883537
```

```
apply(data.1,2,mean)
```

```
##          y         X1         X2      error
## 13.5326971  1.9998118 -0.9314541  0.2050026
```

```
cov(data.1)
```

```
##               y        X1         X2      error
## y     236.91689 17.849920 -17.014278 105.131598
## X1     17.84992  3.189347  -1.031299  -0.675061
## X2    -17.01428 -1.031299   5.202994   1.149705
## error 105.13160 -0.675061   1.149705 111.381164
```

Using this dataset, we can estimate the multiple regression model which corresponding to the true model:

```
lm.out <- lm(y ~ X1 + X2  , data=data.1)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.355  -7.620  -0.422   7.440  32.788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6828     0.7120   2.363   0.0185 *
## X1            4.8502     0.2736  17.727   <2e-16 ***
## X2           -2.3087     0.2142 -10.777   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.56 on 497 degrees of freedom
## Multiple R-squared:  0.5312, Adjusted R-squared:  0.5293
## F-statistic: 281.6 on 2 and 497 DF,  p-value: < 2.2e-16
```

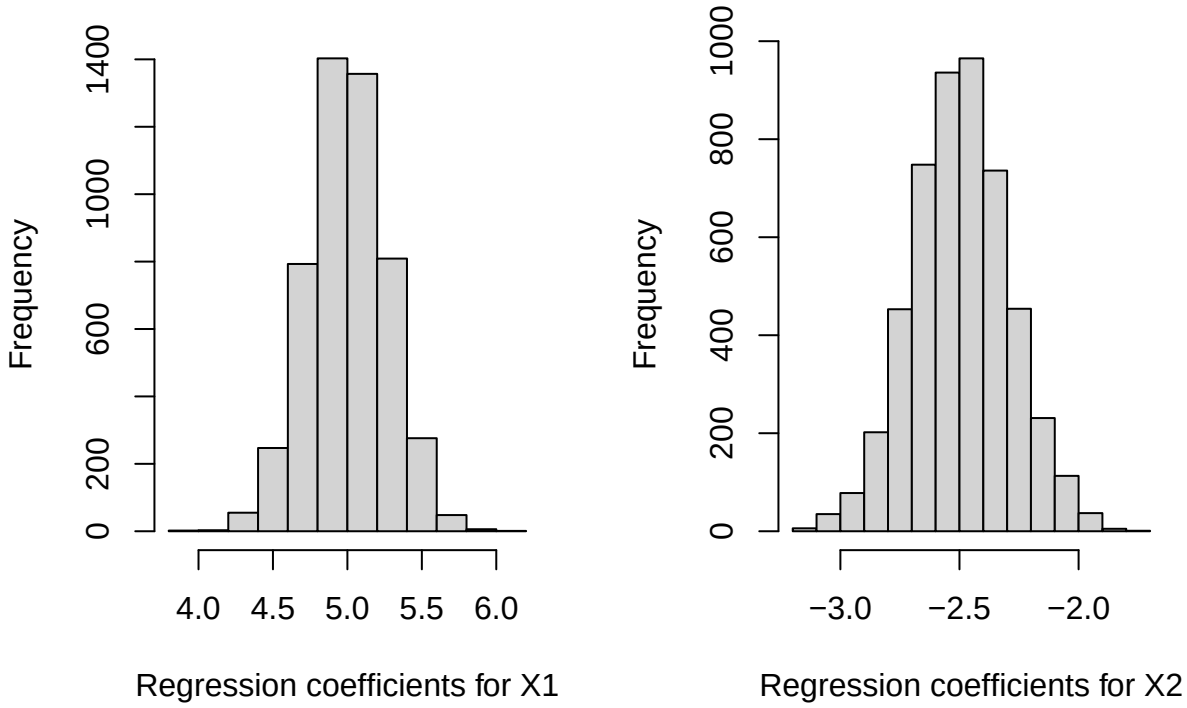
Distribution of the estimates of regression coefficients

We repeat the multiple regression analysis by using each of 5000 datasets:

```
all.coef <- matrix(NA,nrow=num.datasets,ncol=4)
all.coef.sd <- matrix(NA,nrow=num.datasets,ncol=3)

for (i in 1:num.datasets){
    this.data <- GM.samples$generated.data[[i]]

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

    all.coef[i,1:3] <- coef(lm.out)
    all.coef[i,4] <- summary(lm.out)$sigma
    all.coef.sd[i,] <- coef(summary(lm.out))[,2]
}
colnames(all.coef) <- c("b0","b1","b2","sigma")
colnames(all.coef.sd) <- c("b0","b1","b2")
```

Below you will find the distribution of both estimated regression coefficients:

```
par(mfrow=c(1,2))
hist(all.coef[,"b1"],main="",xlab="Regression coefficients for X1")
hist(all.coef[,"b2"],main="",xlab="Regression coefficients for X2")
```



The mean values of these distributions are 5.001 and -2.495. They are almost identical with the true parameter value (5, -2.5). And if we increase the number of generated datasets, we will obtain the identical value with the truth, which means unbiasedness.

Variance of the OLS estimators

According to the textbook, the variance of the OLS slope estimators in a multipre regression model is:

$$Var(\hat{\beta}_j) = \frac{\sigma^2}{SST_j(1 - R_j^2)}$$

. We can check this by using the first generated data.

```
lm.out <- lm(y ~ X1 + X2  , data=data.1)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.355  -7.620  -0.422   7.440  32.788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6828     0.7120   2.363   0.0185 *
## X1            4.8502     0.2736  17.727   <2e-16 ***
```

```
## X2           -2.3087     0.2142 -10.777   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.56 on 497 degrees of freedom
## Multiple R-squared:  0.5312, Adjusted R-squared:  0.5293
## F-statistic: 281.6 on 2 and 497 DF,  p-value: < 2.2e-16
```

For this regression result, we try to reconstruct the standard errors of the slope estimates:

```
SST.1 <- sum((data.1$X1 - mean(data.1$X1))^2)
rsq.1 <- summary(lm(X1 ~ X2 , data=data.1))$r.squared

var.1 <- true.err.var/(SST.1 * (1-rsq.1))

sqrt(var.1)
```

```
## [1] 0.2591091
```

```
SST.2 <- sum((data.1$X2 - mean(data.1$X2))^2)
rsq.2 <- summary(lm(X2 ~ X1 , data=data.1))$r.squared

var.2 <- true.err.var/(SST.2 * (1-rsq.2))

sqrt(var.2)
```

```
## [1] 0.2028649
```

The standard errors calculated based on the above formula is similar, but slightly different from the above result. This is because we used the true error variance for calculation while the above regression result is based on the estimated error variance. We can correspondingly replace the true error variance with the estimated variance (squared residual standard error in the above output).

```
SST.1 <- sum((data.1$X1 - mean(data.1$X1))^2)
rsq.1 <- summary(lm(X1 ~ X2 , data=data.1))$r.squared

var.1 <- (summary(lm.out)$sigma)^2/(SST.1 * (1-rsq.1))

sqrt(var.1)
```

```
## [1] 0.2736112
```

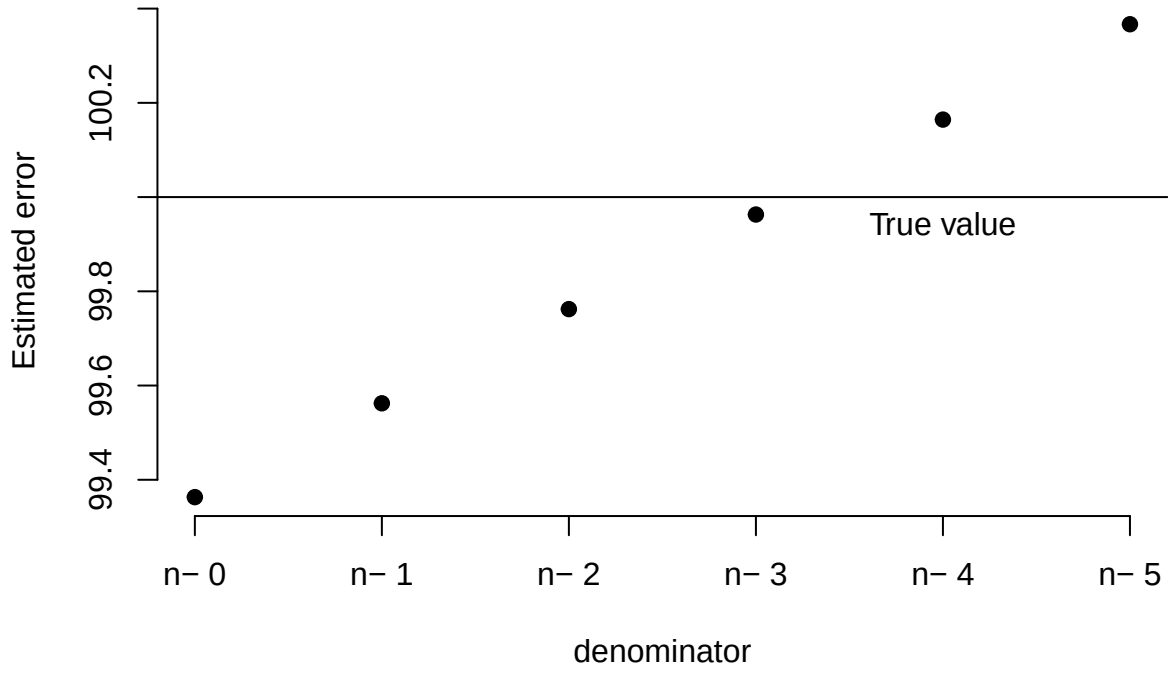
```
SST.2 <- sum((data.1$X2 - mean(data.1$X2))^2)
rsq.2 <- summary(lm(X2 ~ X1 , data=data.1))$r.squared

var.2 <- (summary(lm.out)$sigma)^2/(SST.2 * (1-rsq.2))

sqrt(var.2)
```

```
## [1] 0.214219
```

Here, we have the identical values for the standard errors.

To estimate the error variance, we have to take care about the degrees of freedom. In the simple regression analysis, we divided the sum of square residuals by n-2. The residuals of the above multiple regression model have in contrast n-3 degrees of freedom. Correspondingly, we have to divide the sum of squared residuals by n-3.

```r
all.ssr <- rep(NA,num.datasets)

for (i in 1:num.datasets){
    this.data <- GM.samples$generated.data[[i]]

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

    all.ssr[i] <- sum(lm.out$residuals^2)
}

error.estimates <- NULL
for (i in 0:5){
  error.estimates <- c(error.estimates,
                       mean(all.ssr/(sample.size- i)))
}


plot(error.estimates ~ c(0:5),axes=F,
     xlab="denominator",ylab="Estimated error",pch=19)
axis(1,at=c(0:5),paste("n-",c(0:5)))
axis(2)
abline(h=true.err.var)
text(4,true.err.var,"True value",pos=1)
```



Multicollinearity

The third of the GM-assumptions is slightly different between the simple and multiple regression model. The former assumes only that X has a positive variance. The latter assumes additionally that there is no exact linear relationship among independent variables.

We can see now what will happen if two independent variables of the above multiple regression has an almost linear relationship between both independent variables. This can be generated by setting the correlation of both X with almost one:

```
x.Sigma.mc <- cbind(c(3,-4.55),
                    c(-4.55,7))
cov2cor(x.Sigma.mc)
```

```
##            [,1]       [,2]
## [1,]  1.0000000 -0.9928914
## [2,] -0.9928914  1.0000000
```

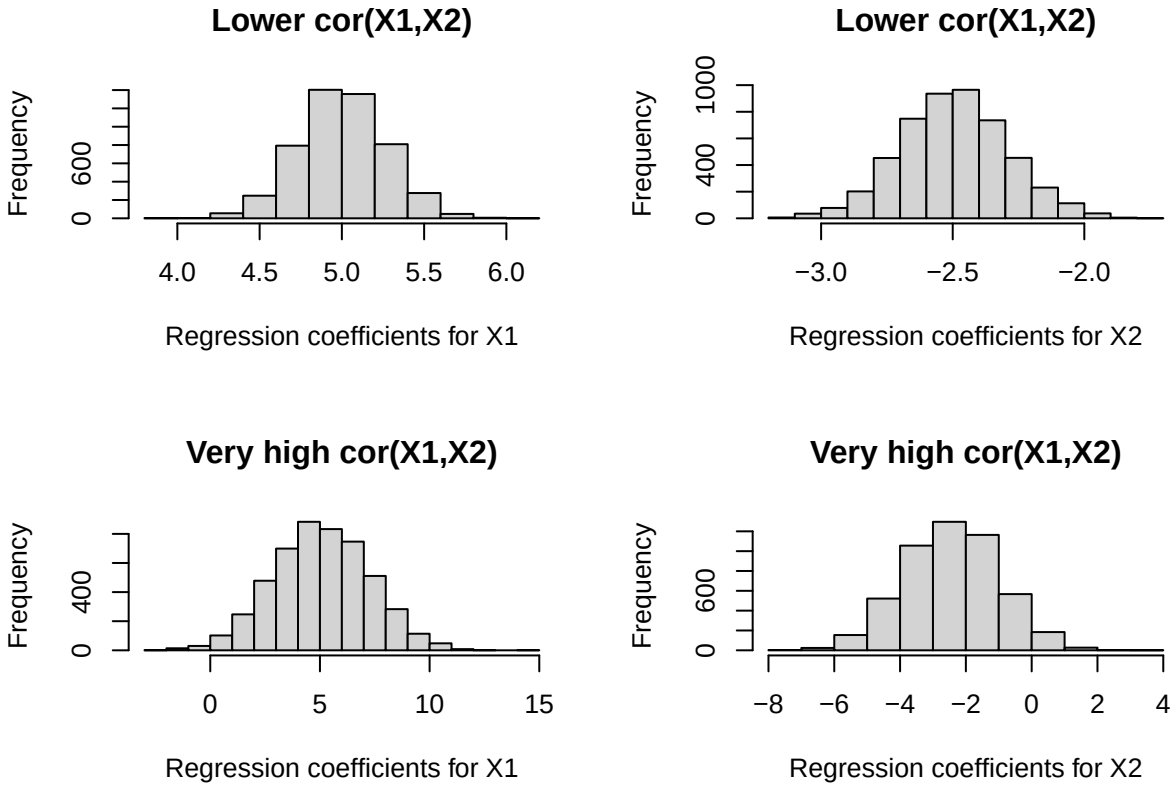
```
MC.samples <- data.generation(sample.size=sample.size,
                              n.sim=num.datasets,
                              n.iv=n.iv,
                              x.mu=x.mu,
                              x.Sigma=x.Sigma.mc,
                              para=c(true.intercept,true.slope),
                              err.dist = "normal",
                              err.disp = true.err.var)
```

Now, we can estimate the multiple regression models for all generated data:

```
all.coef.mc <- matrix(NA,nrow=num.datasets,ncol=4)

for (i in 1:num.datasets){
    this.data <- MC.samples$generated.data[[i]]

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

    all.coef.mc[i,1:3] <- coef(lm.out)
    all.coef.mc[i,4] <- summary(lm.out)$sigma
}
colnames(all.coef.mc) <- c("b0","b1","b2","sigma")
```

Below you will find the distribution of both estimated regression coefficients based on the previous data (with low correlation between X) and those based on the current data with a very high correlation between X:

```
par(mfrow=c(2,2))
hist(all.coef[,"b1"],main="Lower cor(X1,X2)",xlab="Regression coefficients for X1")
hist(all.coef[,"b2"],main="Lower cor(X1,X2)",xlab="Regression coefficients for X2")

hist(all.coef.mc[,"b1"],main="Very high cor(X1,X2)",xlab="Regression coefficients for X1")
hist(all.coef.mc[,"b2"],main="Very high cor(X1,X2)",xlab="Regression coefficients for X2")
```



While there is no bias, the variance for the estimates based on the current data is much higher. If there is the perfect correlation between both independent variables, the variance becomes infinitely large.

Omitted variable bias: Simple case

Above analysis was based on the the regression model which corresponds to the true model with two independent variables. What will happen if we do not consider the second independent variable and estimate the simple regression model:

```
lm.out <- lm(y ~ X1 + X2  , data=data.1)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ X1 + X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.355  -7.620  -0.422   7.440  32.788
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6828     0.7120   2.363   0.0185 *
## X1            4.8502     0.2736  17.727   <2e-16 ***
## X2           -2.3087     0.2142 -10.777   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 10.56 on 497 degrees of freedom
## Multiple R-squared:  0.5312, Adjusted R-squared:  0.5293
## F-statistic: 281.6 on 2 and 497 DF,  p-value: < 2.2e-16
```

```
lm.out.om <- lm(y ~ X1   , data=data.1)
summary(lm.out.om)
```

```
##
## Call:
## lm(formula = y ~ X1, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.121  -7.631  -0.711   7.559  33.494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3403     0.7871   2.973  0.00309 **
## X1            5.5967     0.2937  19.055  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.72 on 498 degrees of freedom
## Multiple R-squared:  0.4217, Adjusted R-squared:  0.4205
## F-statistic: 363.1 on 1 and 498 DF,  p-value: < 2.2e-16
```

The slope of X1 is estimated to be 4.85 in the multiple regression model, while the simple regression model estimates the same slope to be 5.597.

From the textbook, we know the following relationship:

$$\tilde{\beta}_1 = \hat{\beta}_1 + \hat{\beta}_2 \tilde{\delta}_1$$

, where the last delta term is the slope when we regress X2 on X1:

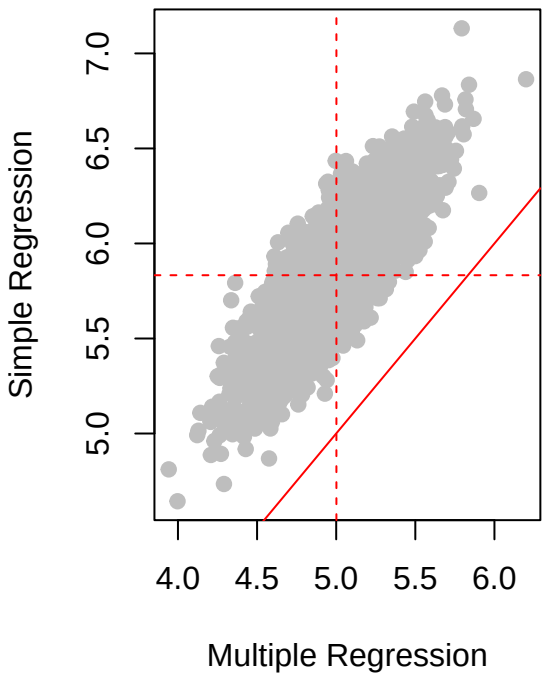
```
lm.out.delta <- lm(X2 ~ X1   , data=data.1)
summary(lm.out.delta)
```

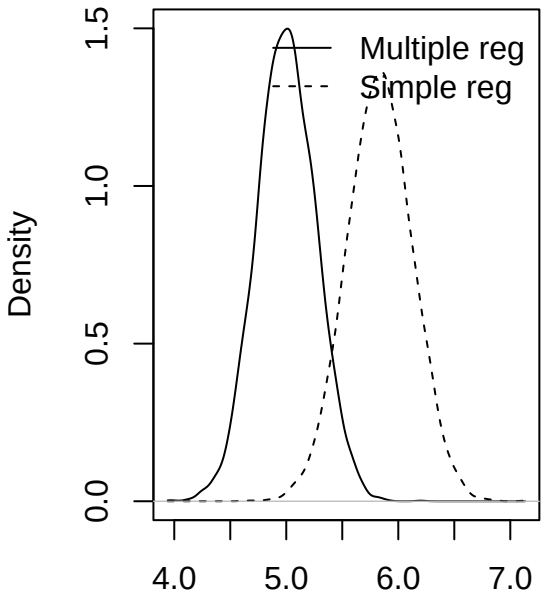
```
##
## Call:
## lm(formula = X2 ~ X1, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7553 -1.3835  0.0383  1.5146  7.4289
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.28480    0.14839  -1.919   0.0555 .
## X1          -0.32336    0.05537  -5.840 9.43e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.209 on 498 degrees of freedom
## Multiple R-squared:  0.06409,    Adjusted R-squared:  0.06221
## F-statistic:  34.1 on 1 and 498 DF,  p-value: 9.434e-09
```

You can check whether the above equation is true by using the numerical examples above.

We can repeat this analysis for all generated data and compare the results:

```
all.coef.om <- matrix(NA,nrow=num.datasets,ncol=3)
all.coef.om.sd <- matrix(NA,nrow=num.datasets,ncol=2)

for (i in 1:num.datasets){
    this.data <- GM.samples$generated.data[[i]]

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

    all.coef.om[i,1:2] <- coef(lm.out)
    all.coef.om[i,3] <- summary(lm.out)$sigma
    all.coef.om.sd[i,] <- coef(summary(lm.out))[,2]
}
colnames(all.coef.om) <- c("b0","b1","sigma")
colnames(all.coef.om.sd) <- c("b0","b1")

par(mfrow=c(1,2))
plot(all.coef[,"b1"],all.coef.om[,"b1"],col="grey",pch=19,
     xlab="Multiple Regression",ylab="Simple Regression")
abline(coef=c(0,1),col="red")
abline(v=mean(all.coef[,"b1"]),col="red",lty=2)
abline(h=mean(all.coef.om[,"b1"]),col="red",lty=2)


whole.data <- c(all.coef[,"b1"],all.coef.om[,"b1"])
plot(density(all.coef[,"b1"],from=min(whole.data),to=max(whole.data)),
     main="",xlab="",ylab="Density",
     ylim=c(0,1.5))
par(new=T)
plot(density(all.coef.om[,"b1"],from=min(whole.data),to=max(whole.data)),
     ann=F,axes=F,
     ylim=c(0,1.5),lty=2)
legend("topright",lty=c(1,2),c("Multiple reg","Simple reg"),bty="n")
```



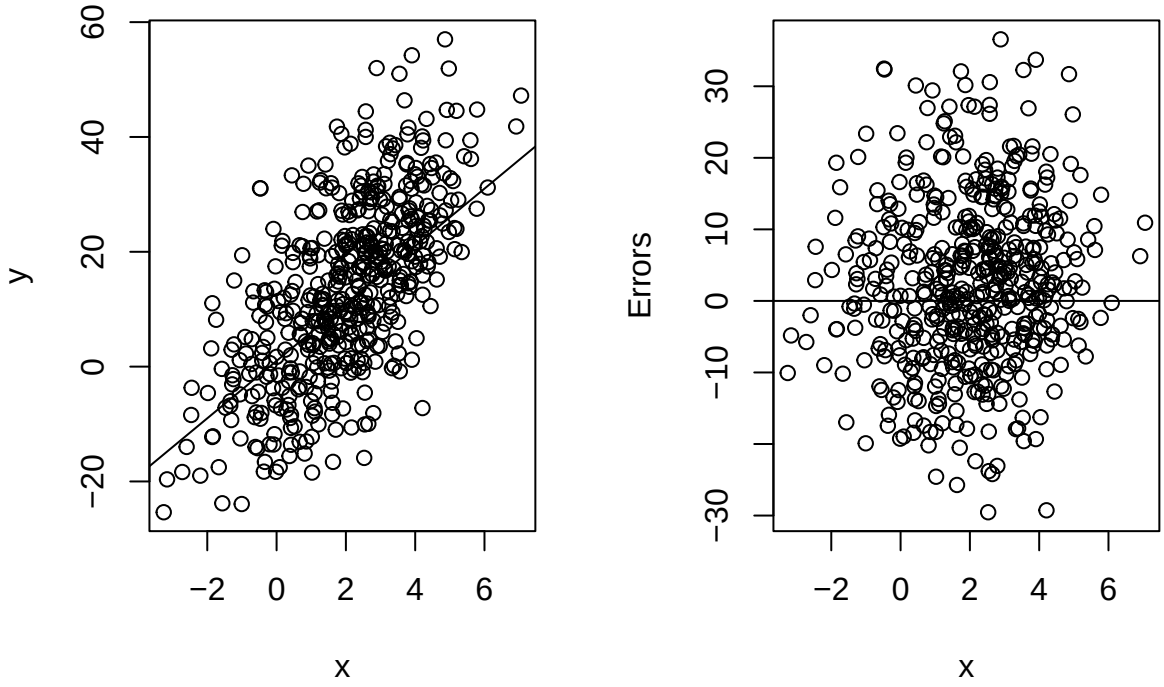


The solid red line corresponds to the 45-degree line. Both dotted red lines corresponds to the mean of each estimates. Accordingly, the simple regression over-estimates the slope of X1. The mean difference of both estimates constitutes the bias caused by the omitted variable (here X2): the omitted variable bias.

The direction of the omitted variable bias is determined by the covariance of both independent variables (-1) and the slope of the omitted variable (-2.5). Check the direction with Table 3.2 of the textbook.

The omitted variable bias is a consequence of violation of the zero conditional mean. To see this, we observe again the first generated dataset:

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

plot(data.1$y ~ data.1$X1,ylab="y",xlab="x")
abline(coef=GM.samples$para[1:2])

error.om <- data.1$error + data.1$X2*true.slope[2]
plot(error.om ~ data.1$X1,ylab="Errors",xlab="x")
abline(h=0)
```

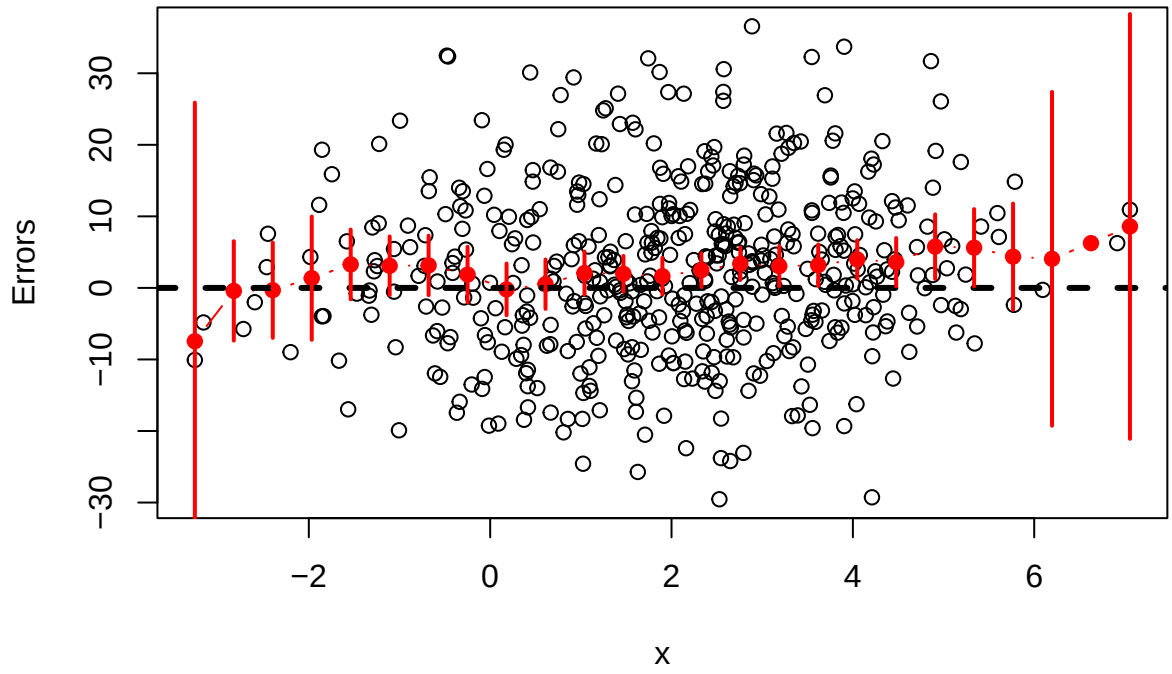


By using the right-hand side panel, we can see whether the zero conditional mean is violated:

```
error.om <- data.1$error + data.1$X2*true.slope[2]
#y.range <- range(data.1$error)
y.range <- range(error.om)
x.range <- range(data.1$X1)

plot(error.om ~ data.1$X1,ylab="Errors",xlab="x",
     xlim=x.range,ylim=y.range)

x.values <- seq(min(data.1$X1),max(data.1$X1),length=25)
x.interval <- x.values[2] -x.values[1]

conditional.mean <- lower.b <- upper.b <- rep(NA,length(x.values))
for (i in 1:length(conditional.mean)){

  selected.error <- error.om[(data.1$X1>(x.values[i]-x.interval)) &
                                          (data.1$X1<(x.values[i]+x.interval)) ]
  conditional.mean[i] <- mean(selected.error)
  this.ci <- ci.sample.mean(selected.error)
  lower.b[i] <- this.ci$lower.b
  upper.b[i] <- this.ci$upper.b

}
```

```
## Warning in qt(bounds.prob, df = length(x) - 1): NaNs wurden erzeugt
```

```
par(new=T)
plot(conditional.mean ~ x.values,ann=F,axes=F,
     xlim=x.range,ylim=y.range,
     col="red",pch=19,type="b")
abline(h=0,lty=2,lwd=3)
for (i in 1:length(conditional.mean)){
  lines(rep(x.values[i],2),c(upper.b[i],lower.b[i]),col="red",lwd=2)
}
```



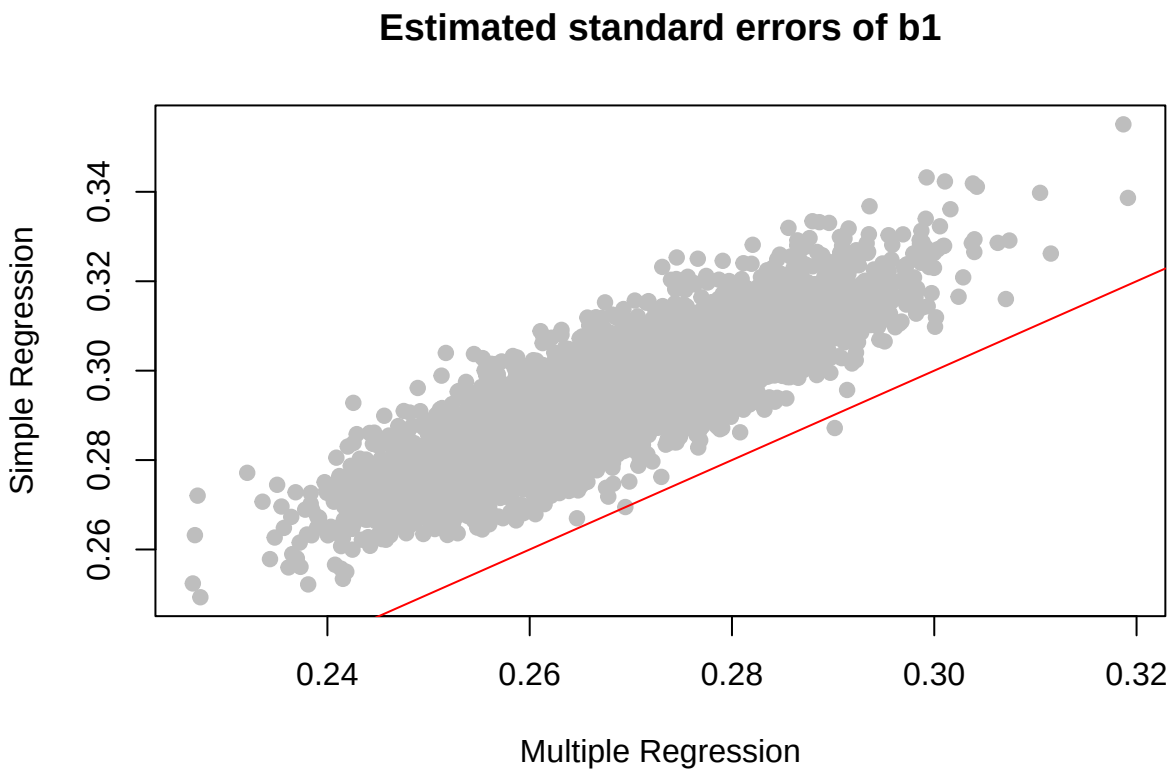
In the figure, it is clearly to see the violation of the zero conditional mean.

Related to the omitted variable bias, there is a statement in the textbook, which may be misleading. According to the statement, the following has to be the case:

$$Var(\tilde{\beta}_1) < Var(\hat{\beta}_1)$$

If we compare the estimated standard errors based on the simulated data, we find the opposite (see Figure below):

```
plot(all.coef.sd[,"b1"],all.coef.om.sd[,"b1"],col="grey",pch=19,
     xlab="Multiple Regression",ylab="Simple Regression",
     main="Estimated standard errors of b1")
abline(coef=c(0,1),col="red")
```



Here, the above inequality is reversed. That is, the biased coefficient has a larger variance than the unbiased coefficient. This difference is because the above textbook's statement is based on the true error variance, while the comparison here is based on the estimated error variance. The estimated error variance is based on that of residuals, which is larger due to the omitted variable (here X2). Actually, Wooldridge does know this fact and states it after the above statement.

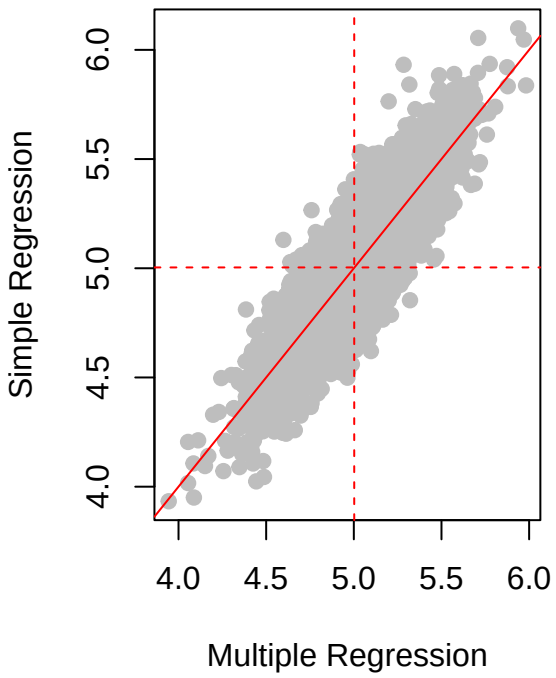
When do we have no omitted variable bias

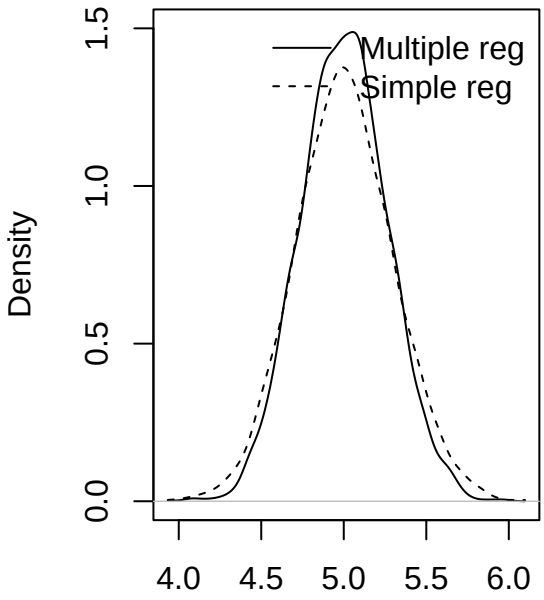
There are two possibilities where we have no omitted variable bias. First, the omitted variable has no impact on y. Second, the omitted variable has zero correlation with the other independent variable. The latter case can be now simulated by setting the corresponding covariance with 0:

```
x.Sigma.2 <- cbind(c(3,0),
                   c(0,5))

GM.samples.2 <- data.generation(sample.size=sample.size,
                                n.sim=num.datasets,
                                n.iv=n.iv,
                                x.mu=x.mu,
                                x.Sigma=x.Sigma.2,
                                para=c(true.intercept,true.slope),
                                err.dist = "normal",
                                err.disp = true.err.var)
```

We can now repeat the regresson analysis with and without the second independent variable as before:

```r
all.coef <- matrix(NA,nrow=num.datasets,ncol=4)

for (i in 1:num.datasets){
    this.data <- GM.samples.2$generated.data[[i]]

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

    all.coef[i,1:3] <- coef(lm.out)
    all.coef[i,4] <- summary(lm.out)$sigma
}
colnames(all.coef) <- c("b0","b1","b2","sigma")


all.coef.om <- matrix(NA,nrow=num.datasets,ncol=3)

for (i in 1:num.datasets){
    this.data <- GM.samples.2$generated.data[[i]]

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

    all.coef.om[i,1:2] <- coef(lm.out)
    all.coef.om[i,3] <- summary(lm.out)$sigma
}
colnames(all.coef.om) <- c("b0","b1","sigma")

par(mfrow=c(1,2))

plot(all.coef[,"b1"],all.coef.om[,"b1"],col="grey",pch=19,
     xlab="Multiple Regression",ylab="Simple Regression")
abline(coef=c(0,1),col="red")
abline(v=mean(all.coef[,"b1"]),col="red",lty=2)
abline(h=mean(all.coef.om[,"b1"]),col="red",lty=2)

whole.data <- c(all.coef[,"b1"],all.coef.om[,"b1"])
plot(density(all.coef[,"b1"],from=min(whole.data),to=max(whole.data)),
     main="",xlab="",ylab="Density",
     ylim=c(0,1.5))
par(new=T)
plot(density(all.coef.om[,"b1"],from=min(whole.data),to=max(whole.data)),
     ann=F,axes=F,
     ylim=c(0,1.5),lty=2)
legend("topright",lty=c(1,2),c("Multiple reg","Simple reg"),bty="n")
```





It is clearly to see both estimates are unbiased.