Chapter 4: Multiple Regression Analysis: Inference

Susumu Shikano

Last compiled at 13. Juli 2022

Classic Linear Model (CLM) assumptions

We generate 5000 datasets with n=25 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 repeat this data generation twice. The first one assumes the normally distributed errors and the second one assumes uniformly distributed errors. In both cases, The variance of the error is 100.

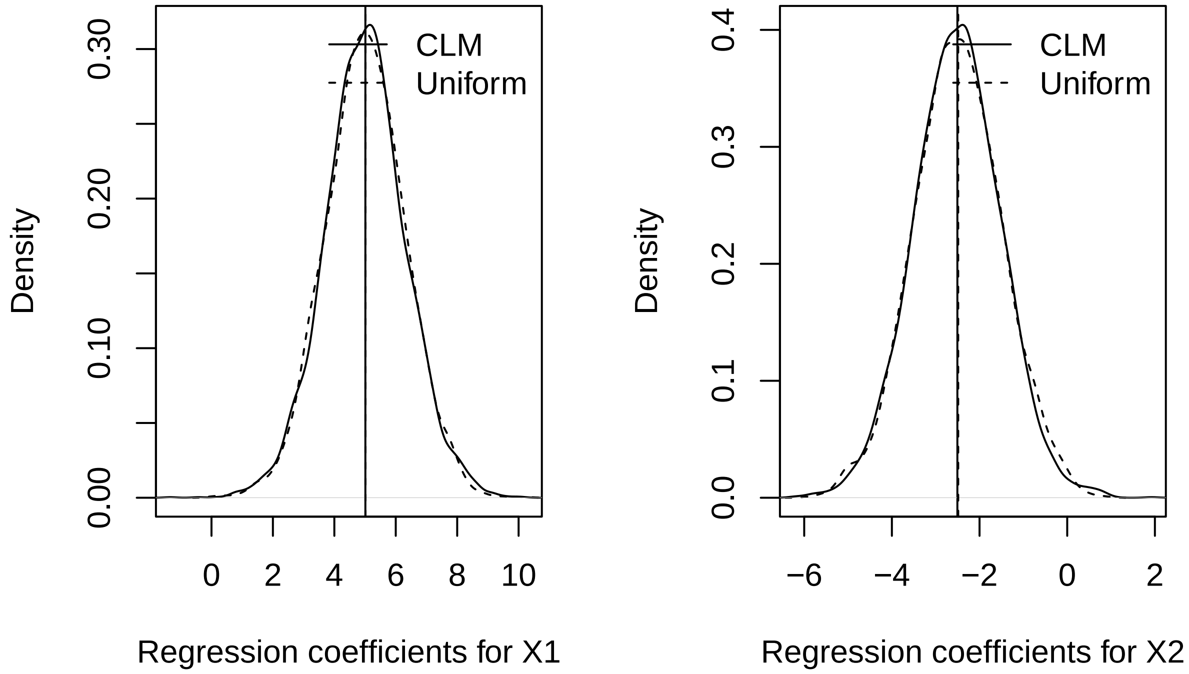
```
CLM.samples <- data.generation(sample.size=sample.size,
                               n.sim=num.datasets,
                               n.iv=n.iv,
                               x.mu=x.mu,
                               x.Sigma=x.Sigma,
                               para=c(true.intercept,true.slope),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

unif.range <- sqrt(true.err.var*12) # transform the variance into the range
nonCLM.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 = "uniform",
                               err.disp = unif.range)
```

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

```
all.coef <- array(NA,dim=c(num.datasets,4,2))
all.coef.se <- array(NA,dim=c(num.datasets,3,2))

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

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

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

    this.data <- nonCLM.samples$generated.data[[i]]
    lm.out <- lm(y ~ X1 + X2 ,data=  this.data)

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



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

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

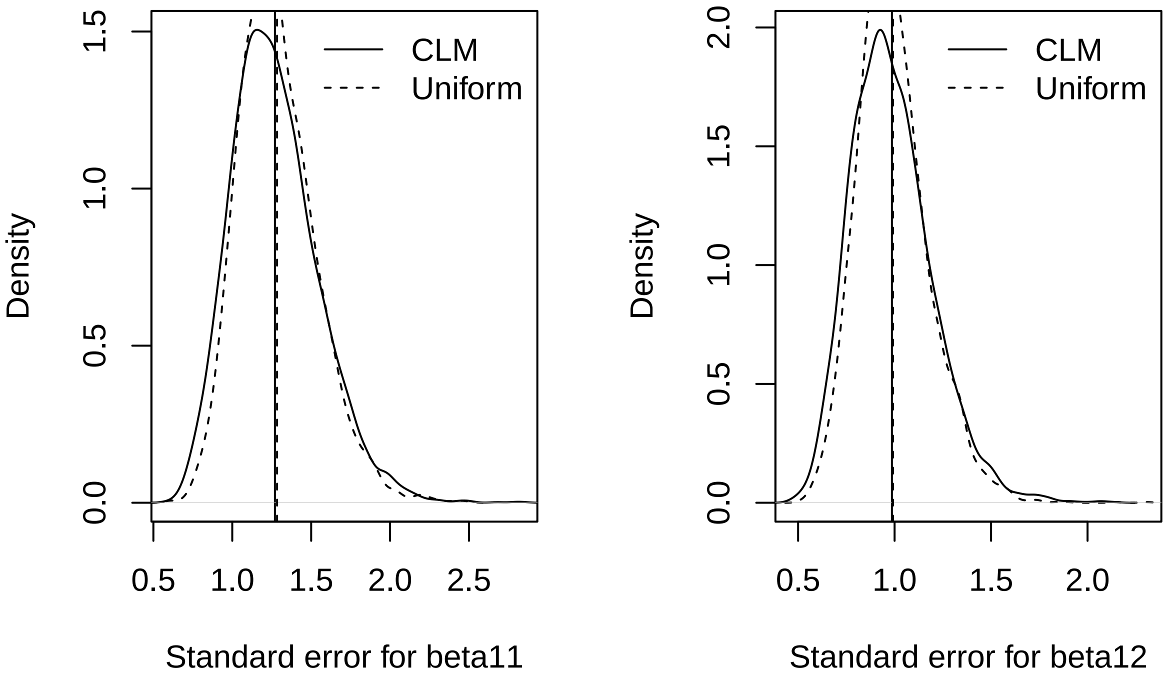
```
par(mfrow=c(1,2))

for (i.fig in 1:2){
    x.range <- range(c(all.coef[,c("b1","b2")[i.fig],]))
    plot(density.out <- density(all.coef[,c("b1","b2")[i.fig],1]),
         main="",xlab=paste0("Regression coefficients for X",i.fig),
         xlim=x.range)
    abline(v=mean(all.coef[,c("b1","b2")[i.fig],1]))
    par(new=T)
    plot(density(all.coef[,c("b1","b2")[i.fig],2]),ann=F,axes=F,
         xlim=x.range,lty=2,
         ylim=c(0,max(density.out$y)))
    abline(v=mean(all.coef[,c("b1","b2")[i.fig],2]),lty=2)
    legend("topright",lty=c(1,2),c("CLM","Uniform"),bty="n")

}
```



It is apparent that estimators are unbiased for both sets of datasets. This also corresponds to the Gauss-Markov theorem since both data generating processes fulfill all the GM-assumptions.

In contrast, the distributions of the estimated standard errors look different as the figure below suggests:

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

for (i.fig in 1:2){
    x.range <- range(c(all.coef.se[,c("b1","b2")[i.fig],]))
    plot(density.out <- density(all.coef.se[,c("b1","b2")[i.fig],1]),
         main="",xlab=paste0("Standard error for beta1",i.fig),
         xlim=x.range)
    abline(v=mean(all.coef.se[,c("b1","b2")[i.fig],1]))
    par(new=T)
    plot(density(all.coef.se[,c("b1","b2")[i.fig],2]),ann=F,axes=F,
         xlim=x.range,lty=2,
         ylim=c(0,max(density.out$y)))
    abline(v=mean(all.coef.se[,c("b1","b2")[i.fig],2]),lty=2)
    legend("topright",lty=c(1,2),c("CLM","Uniform"),bty="n")
}
```



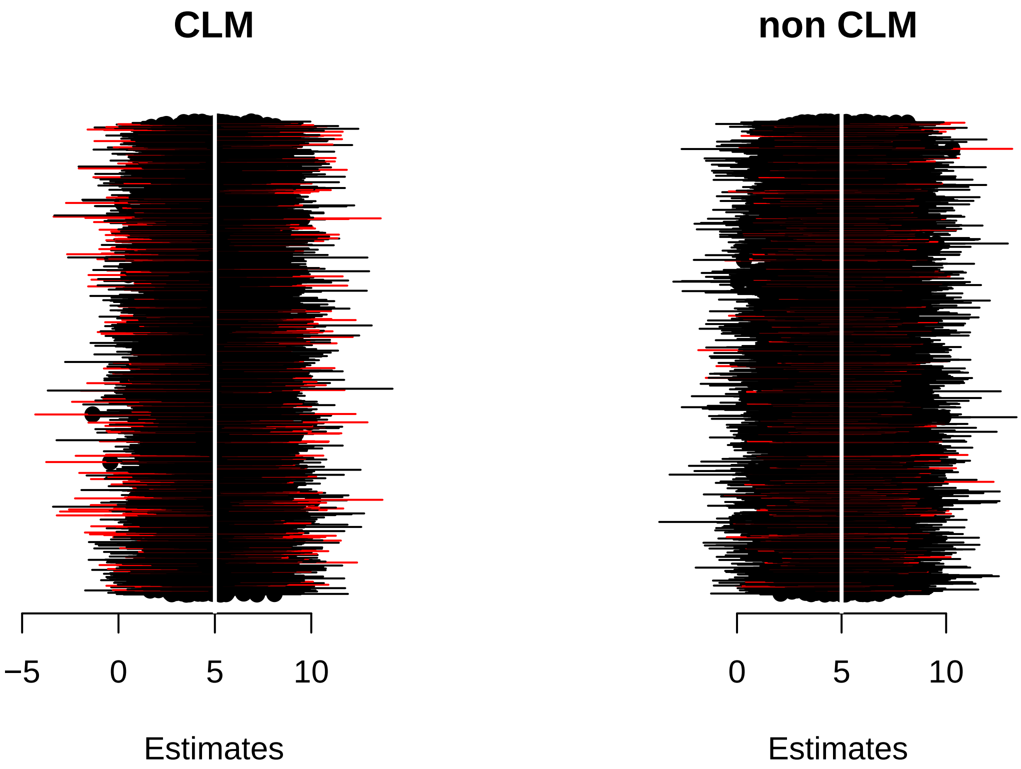
The distributions of the estimated standard errors differ in their shapes among both sets of datasets. Those based on the uniform distribution have less dispertion than thosen based on the normal distribution.

This suggest that the t-values, the point estimates divided by their standard errors, follow different distributions. Since the CLM assumptions guarantees that the above t-values follow a t-distribution, the t-values not based on the CLM assumptions do not follow the t-distribution.

Confidence intervals

We can now construct confidence intervals for all generated samples.

```
include.true <- NULL
par(mfrow=c(1,2))

for (i.fig in 1:2){
    point.e <- all.coef[,"b1",i.fig]
    se <- all.coef.se[,"b1",i.fig]
    true.value <- true.slope[1]

    ci <-  cbind(qt(c(0.025),df=sample.size-3)*se + point.e,
                 qt(c(0.975),df=sample.size-3)*se + point.e)

    range.x <- range(ci)
    include.true <- cbind(include.true,
      ifelse(ci[,1]<true.value & ci[,2]>true.value,1,0))

    plot(point.e,1:length(point.e),pch=19,xlim=range.x,
         main=c("CLM","non CLM")[i.fig],
         axes=F,ylab="",xlab="Estimates"
         )
    axis(1)
    for (i in 1:length(point.e)){
      lines(ci[i,],rep(i,2),col=c("black","red")[2-include.true[i]])
    }
    abline(v=true.slope[1],col="white",lwd=2)

    }
```



For both sets of datasets, about 95% of confidence intervals cover the true population value (94.6%). The confidence intevals based on the data without the CLM-assumptions also perform quite well (95.26%).

Effect of scales on the regression coefficients (revisited)

Now we can generate datasets with 2 independent variables whose slope are very similar.

```
CLM.samples.2 <- data.generation(sample.size=sample.size,
                                 n.sim=1000,
                                 n.iv=n.iv,
                                 x.mu=x.mu,
                                 x.Sigma=x.Sigma,
                                 para=c(true.intercept,c(5,5.1)),
                                 err.dist = "normal",
                                 err.disp = true.err.var)
```

If we estimate the multiple regression model with the first generated data:

```
data.1 <- CLM.samples.2$generated.data[[1]]

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

```
##
## Call:
## lm(formula = y ~ X1 + X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.400  -8.192   1.363   7.186  17.610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8293     2.8650   0.289 0.774952
## X1            4.1535     1.2885   3.224 0.003909 **
## X2            3.9017     0.8709   4.480 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.09 on 22 degrees of freedom
## Multiple R-squared:  0.5278, Adjusted R-squared:  0.4849
## F-statistic:  12.3 on 2 and 22 DF,  p-value: 0.0002601
```

The estimated effect size is similar. Now we can test whether their difference is significant. For this purpose, we can construct a new variable by adding both independent variables and replace one of the independent variables with the new variable.

```
X12 <- data.1$X1 +data.1$X2

summary(lm(y ~ X12 + X2 ,data=  data.1))
```

```
##
## Call:
## lm(formula = y ~ X12 + X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.400  -8.192   1.363   7.186  17.610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8293     2.8650   0.289  0.77495
## X12           4.1535     1.2885   3.224  0.00391 **
## X2           -0.2518     1.3535  -0.186  0.85410
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.09 on 22 degrees of freedom
## Multiple R-squared:  0.5278, Adjusted R-squared:  0.4849
## F-statistic:  12.3 on 2 and 22 DF,  p-value: 0.0002601
```

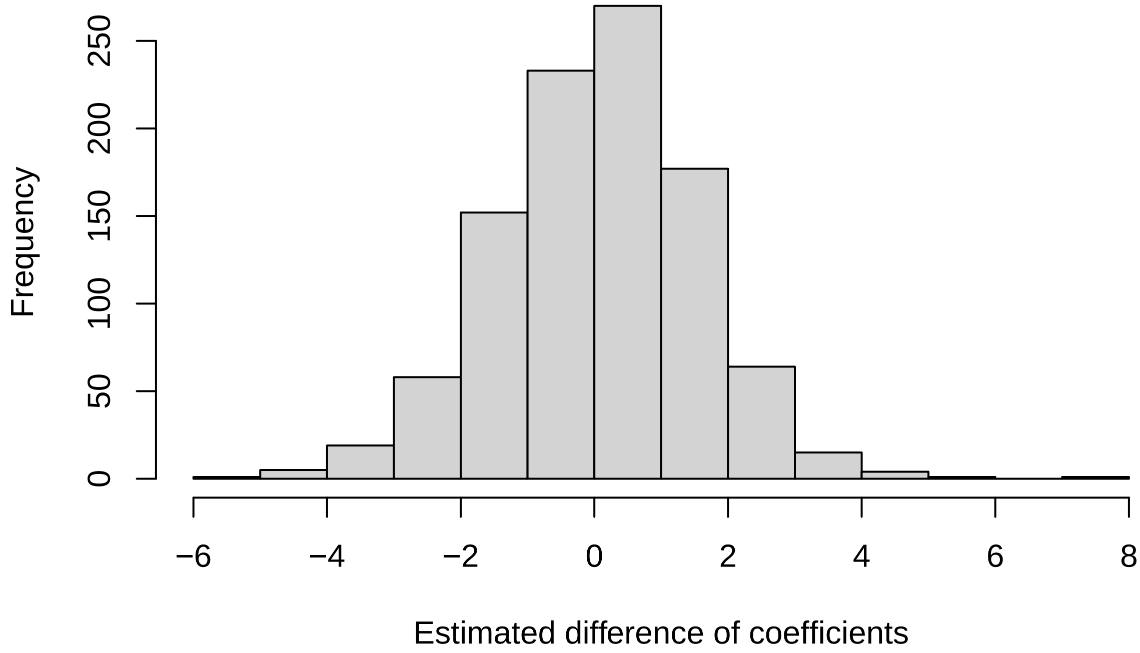
The estimated slope of the new variable (X12) corresponds to the slope of the replaced variable (X1) and the slope of the other independent variable (X2) corresponds to the difference.

We repeat the same analysis for all generated datasets.

```r
all.coef.2 <- array(NA,dim=c(CLM.samples.2$n.sim,4,2))
all.coef.p.2 <- array(NA,dim=c(CLM.samples.2$n.sim,3,2))

for (i in 1:CLM.samples.2$n.sim){
    this.data <- CLM.samples.2$generated.data[[i]]

    X12 <- this.data$X1 +this.data$X2
    lm.out <- lm(y ~ X12 + X2 ,data=  this.data)

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

    X12 <- this.data$X1 +this.data$X2/10
    lm.out <- lm(y ~ X12 + X2 ,data=  this.data)

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

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

The distribution of the estimated difference of both slopes appears below.

```r
hist(all.coef.2[,"b2",1],
     main="",xlab="Estimated difference of coefficients")
```



This distribution's mean is 0.052, which is almost identical with the difference in the true parameters. The p-value for the correct direction of the difference (i.e. the diference is positive) is under 5% only in 5.6% of the datasets.

We can repeat the same exercise with a rescaled X2. More specifically, we create the new X2 variable by dividing X2 and check the difference of slopes.

```
data.1$new.X2 <- data.1$X2/10
summary(lm(y ~ X1 + new.X2 ,data=  data.1))
```

```
##
## Call:
## lm(formula = y ~ X1 + new.X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.400  -8.192   1.363   7.186  17.610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8293     2.8650   0.289 0.774952
## X1            4.1535     1.2885   3.224 0.003909 **
## new.X2       39.0168     8.7089   4.480 0.000187 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.09 on 22 degrees of freedom
## Multiple R-squared:  0.5278, Adjusted R-squared:  0.4849
## F-statistic:  12.3 on 2 and 22 DF,  p-value: 0.0002601
```

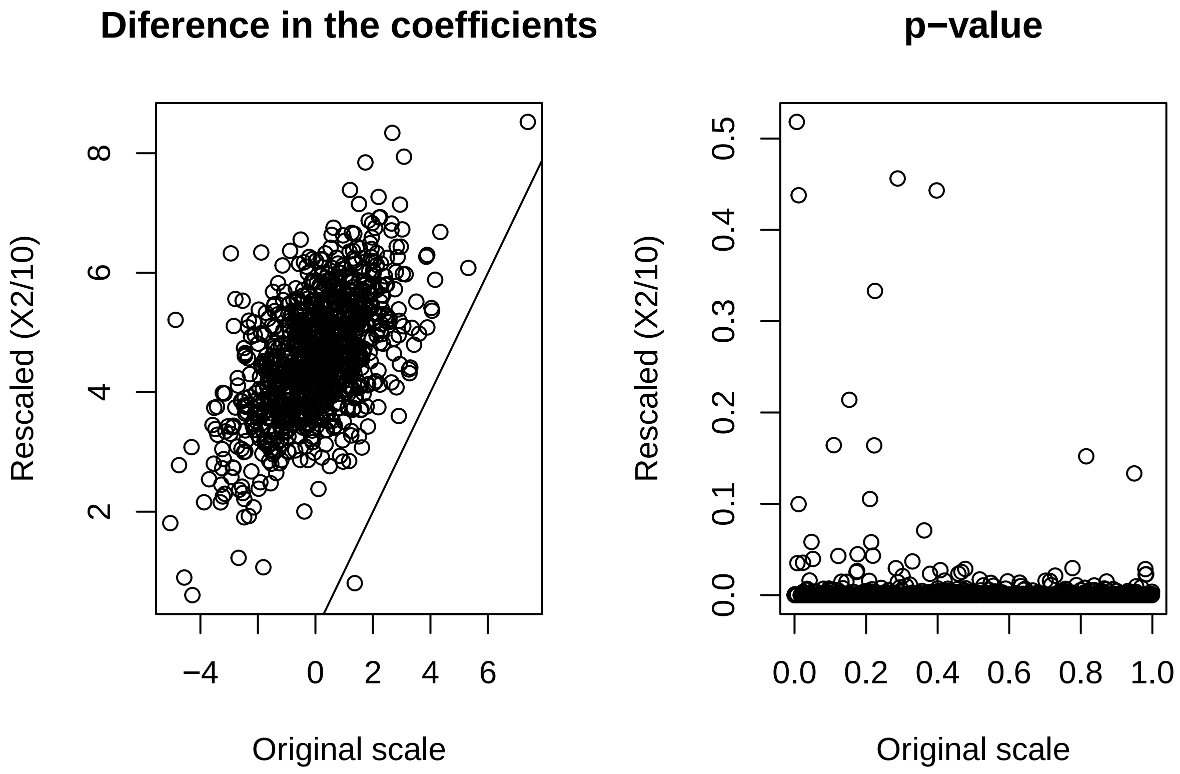
```r
data.1$new.X12 <- data.1$X1 +data.1$new.X2

summary(lm(y ~ new.X12 + new.X2 ,data=  data.1))
```

```
##
## Call:
## lm(formula = y ~ new.X12 + new.X2, data = data.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.400  -8.192   1.363   7.186  17.610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8293     2.8650   0.289 0.774952
## new.X12       4.1535     1.2885   3.224 0.003909 **
## new.X2       34.8633     8.4640   4.119 0.000451 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.09 on 22 degrees of freedom
## Multiple R-squared:  0.5278, Adjusted R-squared:  0.4849
## F-statistic:  12.3 on 2 and 22 DF,  p-value: 0.0002601
```

Since we have a smaller unit for X2, its slope becomes larger. Consequently, the difference between slopes becomes larger.

We can do the same analysis for all the generated datasets and compare the estimated differences in the slopes and their p-values.

```r
par(mfrow=c(1,2))
plot(all.coef.2[,"b2",],
     xlab="Original scale",ylab="Rescaled (X2/10)",
     main="Diference in the coefficients"
     )
abline(coef=c(0,1))

plot(all.coef.p.2[,"b2",],
     xlab="Original scale",ylab="Rescaled (X2/10)",
     main="p-value"
)
```



It is obvious that the estimated difference is larger in all datasets, and p-value is much more often under 5%.

From this exercise, we can learn that the sigificance test of differences in slopes strongly depends on the scale of the independent variables at stake. And such comparison is only meaningful if both variables have the same scale.

Testing against the null hypothesis that multiple independent variables have no effect on Y.

We generate another set of datasets under the CLM-assumptions with no effect for the second and third independent variables.

```
CLM.samples.3 <- data.generation(sample.size=sample.size,
                                 n.sim=1000,
                                 n.iv=3,
                                 x.mu=c(2,-1,3),
                                 x.Sigma=cbind(c(3,-1,1),
                                               c(-1,5,2),
                                               c(1,2,2)),
                                 para=c(true.intercept,c(5,0,0)),
                                 err.dist = "normal",
                                 err.disp = true.err.var)
```

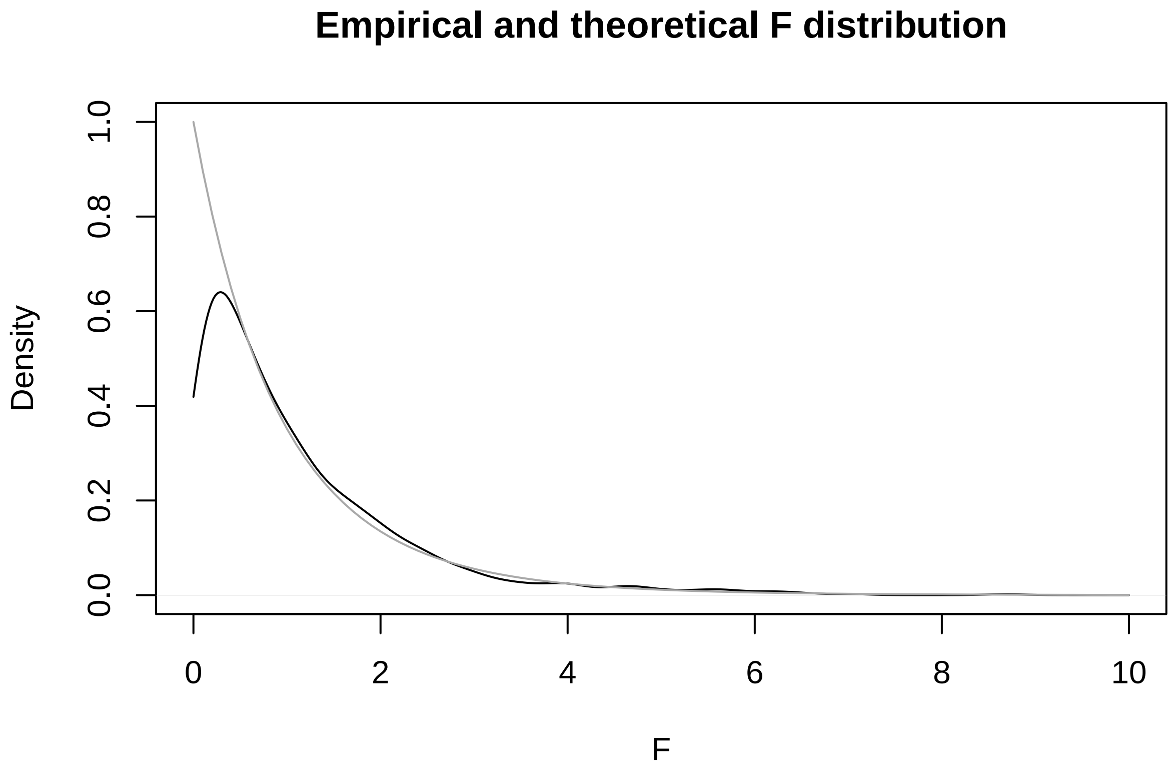
To test against the null hypothesis that the second and third independent variables have no effect on Y, we can rely on the F-test.

We can calculate the F-values for all datasets.

```
all.F<- rep(NA,CLM.samples.3$n.sim)

for (i in 1:CLM.samples.3$n.sim){
    this.data <- CLM.samples.3$generated.data[[i]]

    lm.out <- lm(y ~ X1 + X2 + X3, data=this.data)
    lm.out.res <- lm(y ~ X1 , data=this.data)

    SSR.ur <- sum(lm.out$residuals^2)
    SSR.r <-  sum(lm.out.res$residuals^2)
    q <- lm.out.res$df.residual - lm.out$df.residual

    all.F[i] <-  ((SSR.r - SSR.ur)/q)/(SSR.ur/lm.out$df.residual)

}
```

We can observe the calculated empirical distribution of the F values, which is based on the population model with no effect of X2 and X3.

```
plot(density(all.F,from=0,to=10),xlim=c(0,10),ylim=c(0,1),
     main="Empirical and theoretical F distribution",
     xlab="F")

this.f.func <- function(x) df(x ,q,lm.out$df.residual)
curve(this.f.func,0,10,add=TRUE,col="darkgrey")
```



The empirical distribution is well fitted to the theoretical F-distribution with the corresponding degrees of freedom.