Chapter 7: Multiple Regression Analysis: Dummy Variables

Susumu Shikano

Last compiled at 18. Juli 2022

Linear probability model

We generate 1000 datasets with n=200 under the GM-assumptions. The number of independent variables is 1. The true regression line has the intercept of 0.4 and the slope of 0.1. The independent variables are generated with the mean 1, variances 0.5.

In previous examples, we generated the dependent variable by adding random errors to the predicted values generated above. Here, instead, we use the predicted values as probability that the dependent variable has the value 1. With the opposite probability, the dependent variable has the value 0. This is called Bernoulli trial.

```r
samples.1 <- 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,
                             binary.y = TRUE)
```

Analogously, we generate further two sets of samples. The second set is generated under the same parameters except that the mean value of X is set to 5 instead of 1. The third set is generated under the same parameters of the first set except that the true slope is 0.5 instead of 0.1.

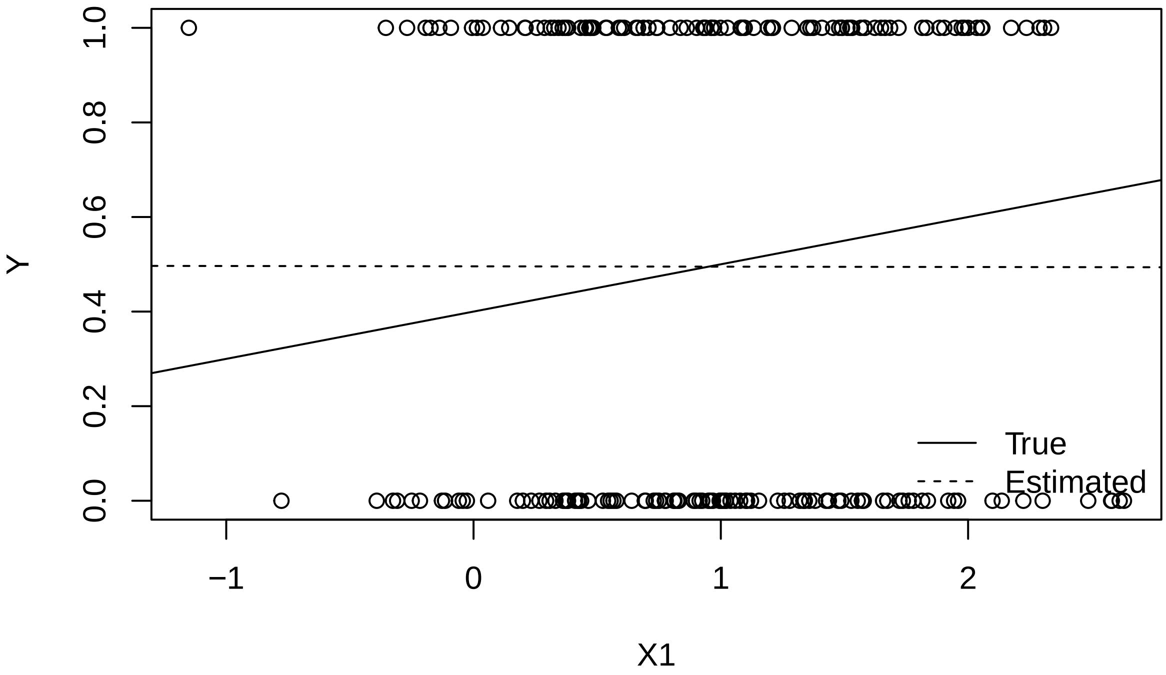
```r
samples.2 <- data.generation(sample.size=sample.size,
                             n.sim=num.datasets,
                             n.iv=n.iv,
                             x.mu=x.mu+4,
                             x.Sigma=x.Sigma,
                             para=c(true.intercept,true.slope),
                             err.dist = "normal",
                             err.disp = true.err.var,
                             binary.y = TRUE)

samples.3 <- 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+0.4),
                             err.dist = "normal",
                             err.disp = true.err.var,
                             binary.y = TRUE)
```

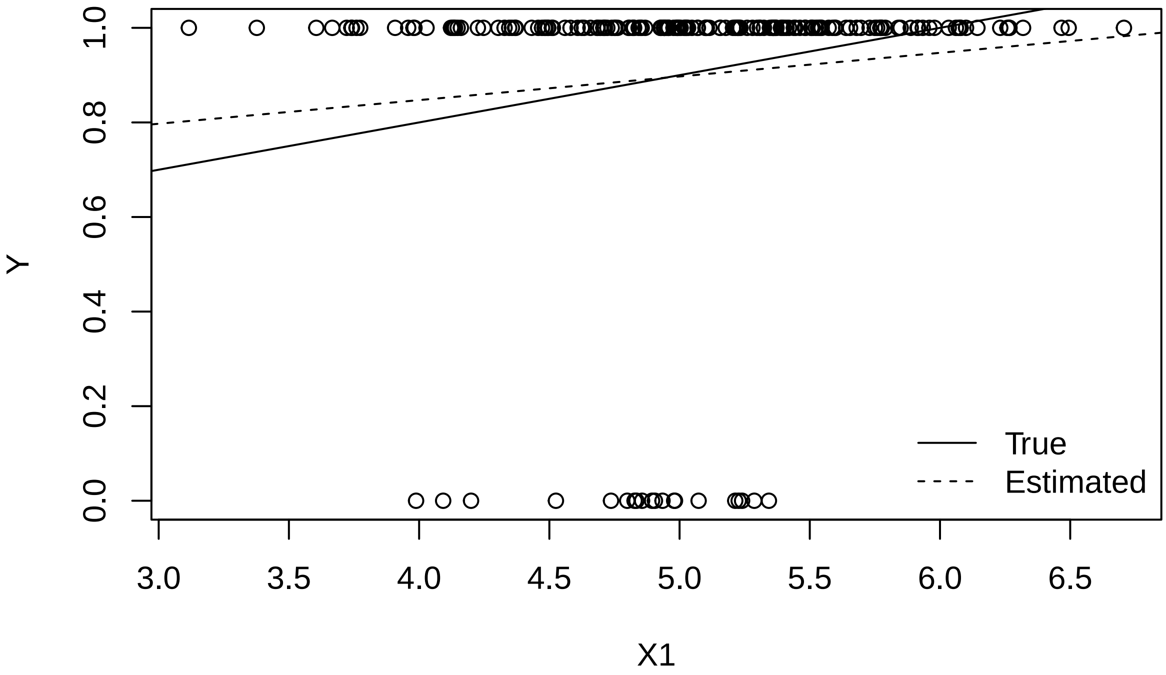
We can estimate the regression model

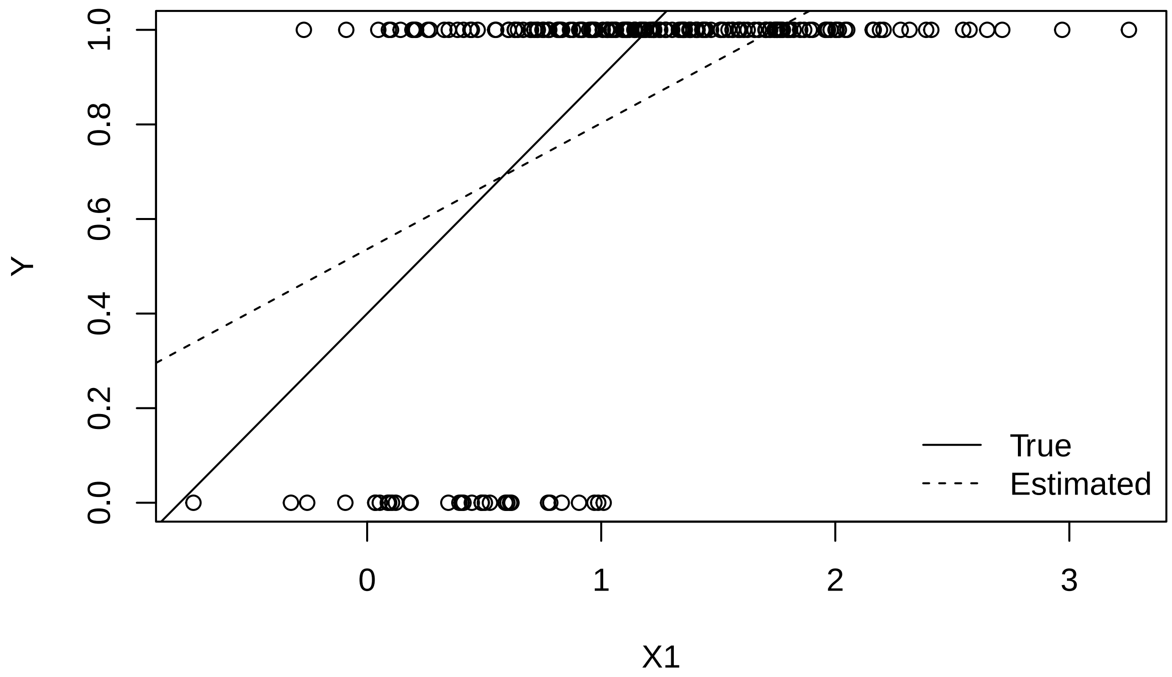
$$\hat{y} = \tilde{\beta}_0 + \tilde{\beta}_1 x_1$$

.

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

for (i.set in 1:3 ){
  if (i.set ==1 ) this.generated.data <- samples.1
  if (i.set ==2 ) this.generated.data <- samples.2
  if (i.set ==3 ) this.generated.data <- samples.3
for (i in 1:num.datasets){
    this.data <- this.generated.data$generated.data[[i]]

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

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

    all.predict.outside[i,i.set,1] <- sum(this.data$y.hat==0|this.data$y.hat==1)
    all.predict.outside[i,i.set,2] <- sum(lm.out$fitted.values<0|lm.out$fitted.values>1)

    # visual presentation
    if (i ==1){
      plot(this.data$y ~ this.data$X1,
           xlab="X1",ylab="Y",main="")
      abline(coef= this.generated.data$para)

      abline(reg=lm.out,lty=2)
      legend("bottomright",lty=c(1,2),
             c("True","Estimated"),
             bty="n")
    }

}
}
```


1.0
0.8
0.6
0.4
0.2
0.0
Y
−1
0
1
2
X1
True
Estimated

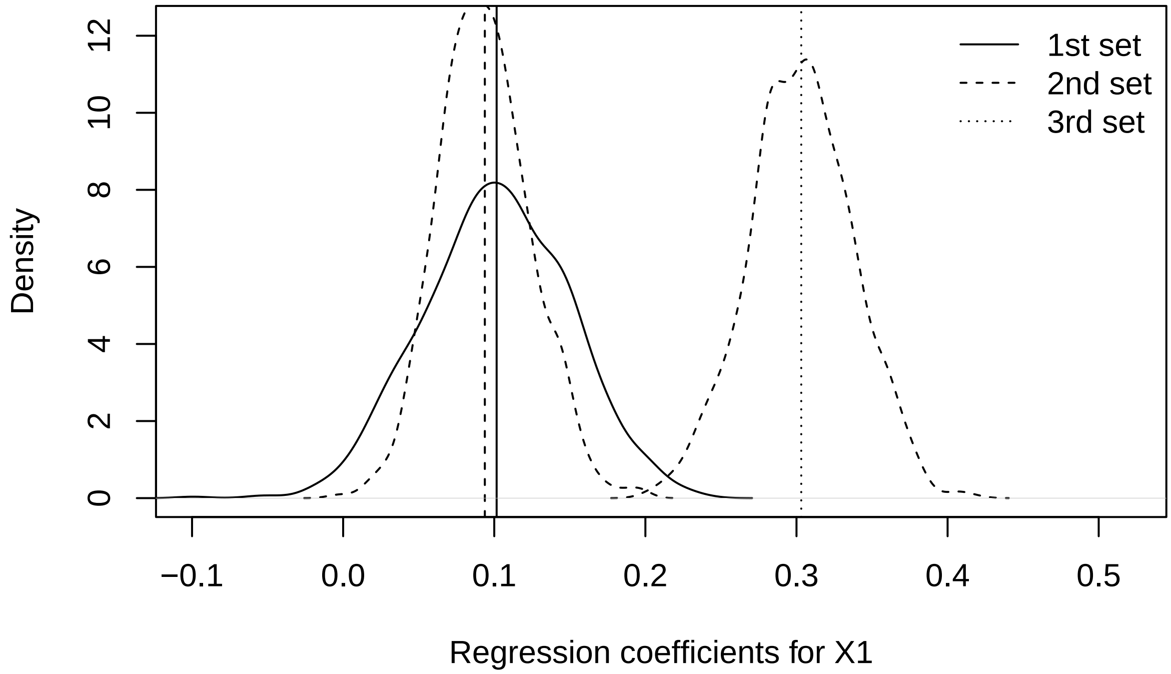

Y
1.0
0.8
0.6
0.4
0.2
0.0
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
X1
True
Estimated



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

Below you will find the distribution of estimated regression coefficients:

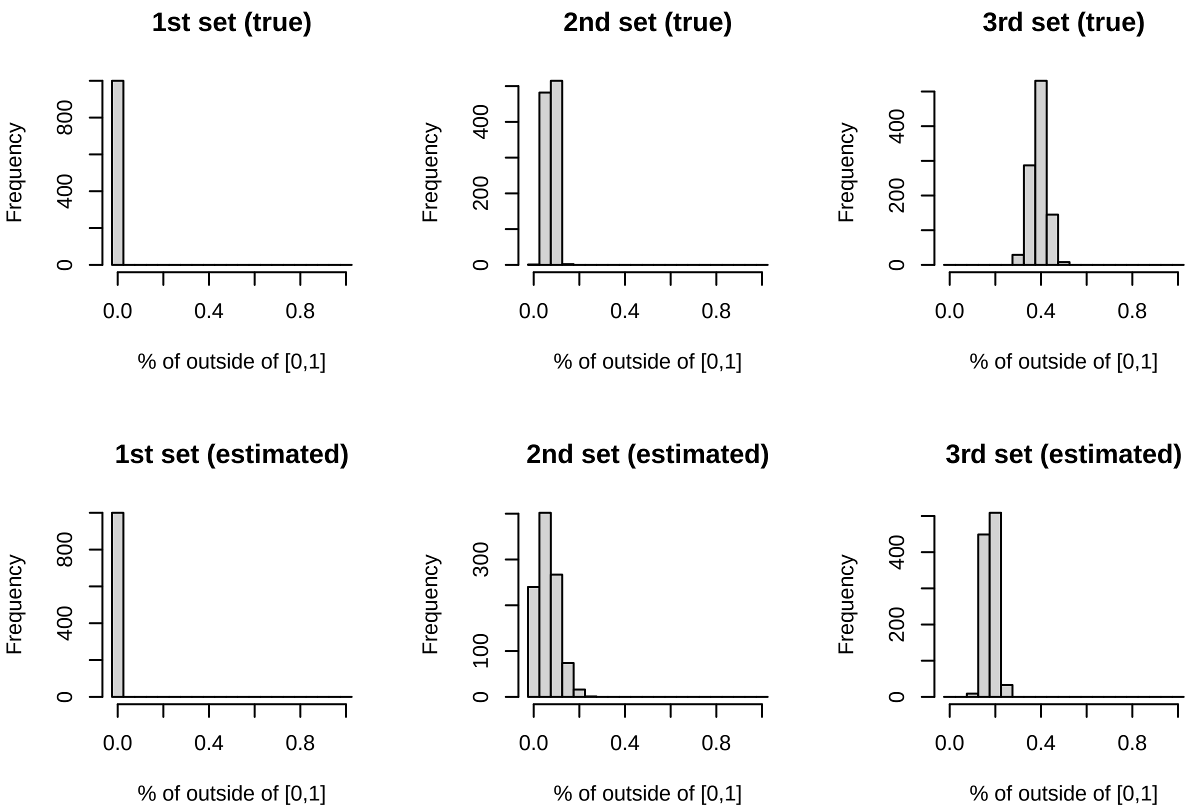
```
    x.range <- range(c(all.coef[,"b1",]))
    x.range[2] <- 0.52
    density.out <- density(all.coef[,"b1",1])
    plot(density.out,
         main="",xlab=paste0("Regression coefficients for X1"),
         xlim=x.range,ylim=c(0,max(density.out$y)*1.5))
    abline(v=mean(all.coef[,"b1",1]))
    par(new=T)
    plot(density(all.coef[,"b1",2]),ann=F,xlab="",ylab="",main="",
         axes=F,
         xlim=x.range,lty=2,
         ylim=c(0,max(density.out$y)*1.5))
    abline(v=mean(all.coef[,"b1",2]),lty=2)
    par(new=T)
    plot(density(all.coef[,"b1",3]),ann=F,xlab="",ylab="",main="",
         axes=F,
         xlim=x.range,lty=2,
         ylim=c(0,max(density.out$y)*1.5))
    abline(v=mean(all.coef[,"b1",3]),lty=3)
    legend("topright",lty=c(1:3),c("1st set","2nd set","3rd set"),bty="n")
```



For the first and second set, the true regression slope is 0.1 while it is set to 0.5 for the third set. Obviously, the estimates based on the second and third set of samples are downwards biased, while those based on the first set are almost bias free.

These results seem to have to do with how often the predicted values are outside of the range [0,1].

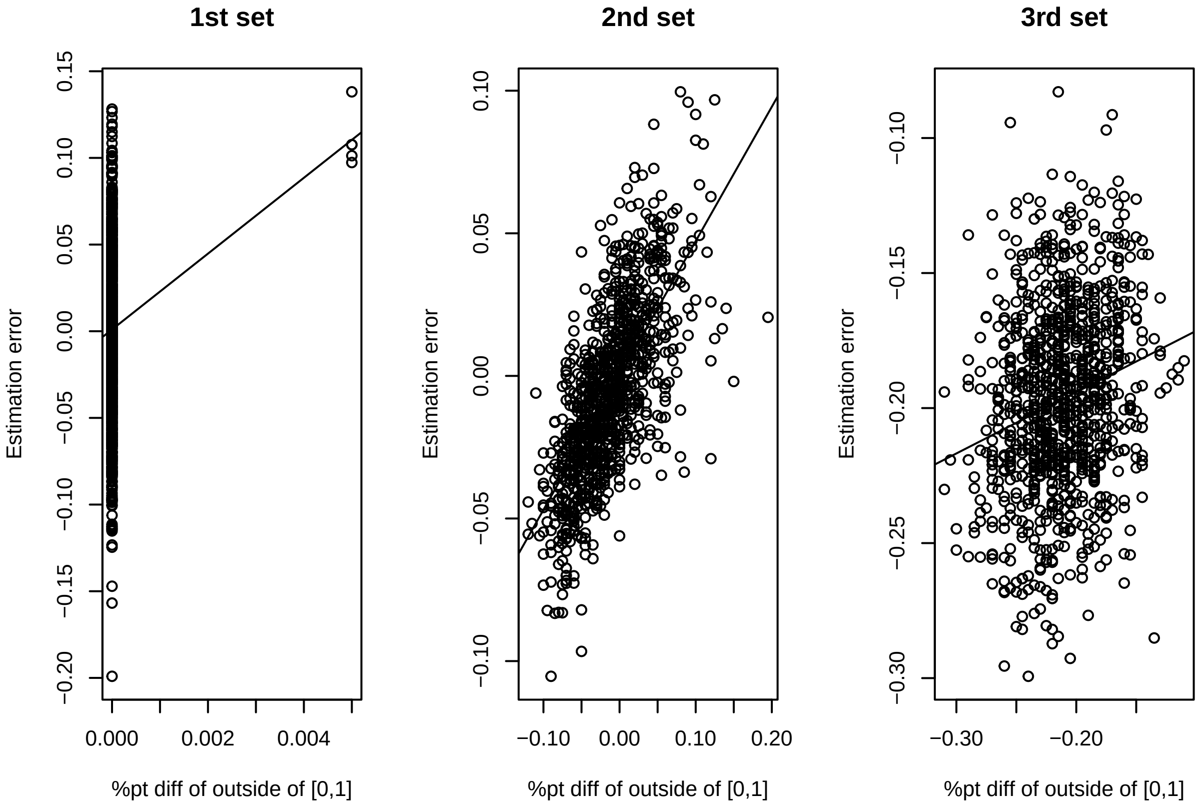
```
par(mfcol=c(2,3))

for (i.fig in 1:3){
  hist(all.predict.outside[,i.fig,1]/sample.size,br=seq(-0.025,1.025,by=0.05),
       xlab="% of outside of [0,1]",
       main=paste(c("1st","2nd","3rd")[i.fig],"set (true)"))

  hist(all.predict.outside[,i.fig,2]/sample.size,br=seq(-0.025,1.025,by=0.05),
       xlab="% of outside of [0,1]",
       main=paste(c("1st","2nd","3rd")[i.fig],"set (estimated)"))

  }
```



The above figure shows in the upper panels how often the predicted values are outside of [0,1] in the data generation process. The lower panels presents, in contrast, how often the predicted values based on the estimated regression model are outside of the range.

These figures are however not crucial for the above bias and estimation error. More important is the difference of the share of predicted values outside of the range between the true model and estimated results. These differences are plotted against the estimation error (estimated slope minus the true slope) in the figure below.

```r
par(mfrow=c(1,3))
for (i.fig in 1:3){
  if (i.fig <= 2){
      this.error <- all.coef[,2,i.fig] - true.slope
  } else {
      this.error <- all.coef[,2,i.fig] - true.slope-0.4
  }

  this.share <- (all.predict.outside[,i.fig,2]-all.predict.outside[,i.fig,1])/sample.size

  plot(this.share,this.error,
       xlab="%pt diff of outside of [0,1]",
       ylab="Estimation error",
       main=paste(c("1st","2nd","3rd")[i.fig],"set"))
  abline(lm(this.error ~ this.share))
}
```



The first two panels demonstrate that zero difference in the share of predicted values outside of the range is associated with the unbiased results.