Chapter 2: Simple Regression Model

Susumu Shikano

Last compiled at 13. Juli 2022

Simple Regression under the GM-assumptions

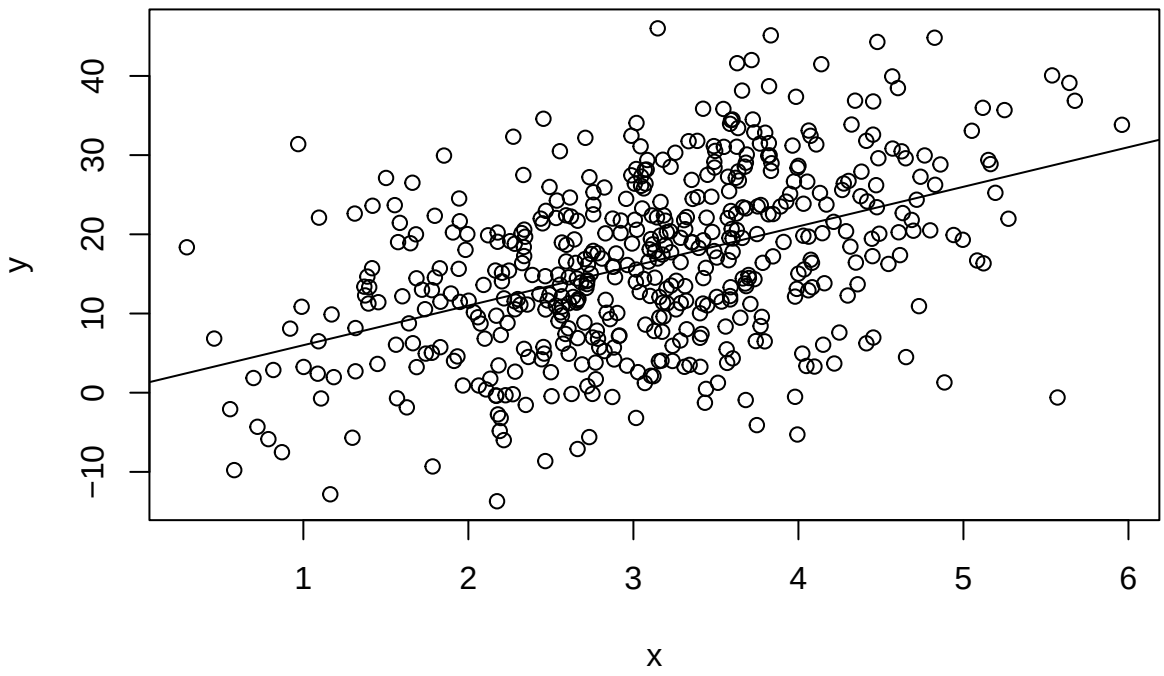
We generate 5000 datasets with n=500 under the GM-assumptions. The true regression line has the intercept of 1 and the slope of 5. The variance of the error is 100.

```
GM.samples <- data.generation(sample.size=sample.size,
                              n.sim=num.datasets,
                              n.iv=1,
                              x.mu=3,
                              x.Sigma=as.matrix(1,nrow=1),
                              para=c(true.intercept,true.slope),
                              err.dist = "normal",
                              err.disp = true.err.var)
```

Describing a generated dataset and checking the zero conditional mean assumption

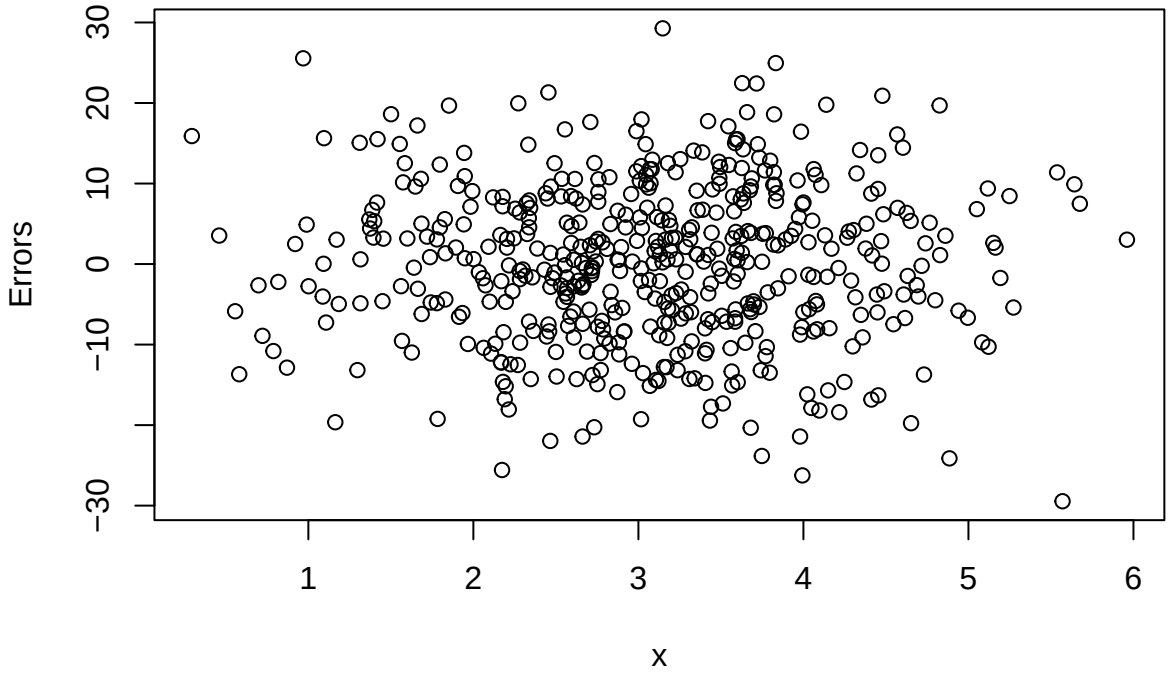
The first generated dataset looks as follows:

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

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



The line through the data is the true regression line, based on which the dataset was generated. By subtracting the true y from the observed y, we can obtain errors (which has been stored during the data generation above):

```
y.range <- range(data.1$error)
x.range <- range(data.1$X1)

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

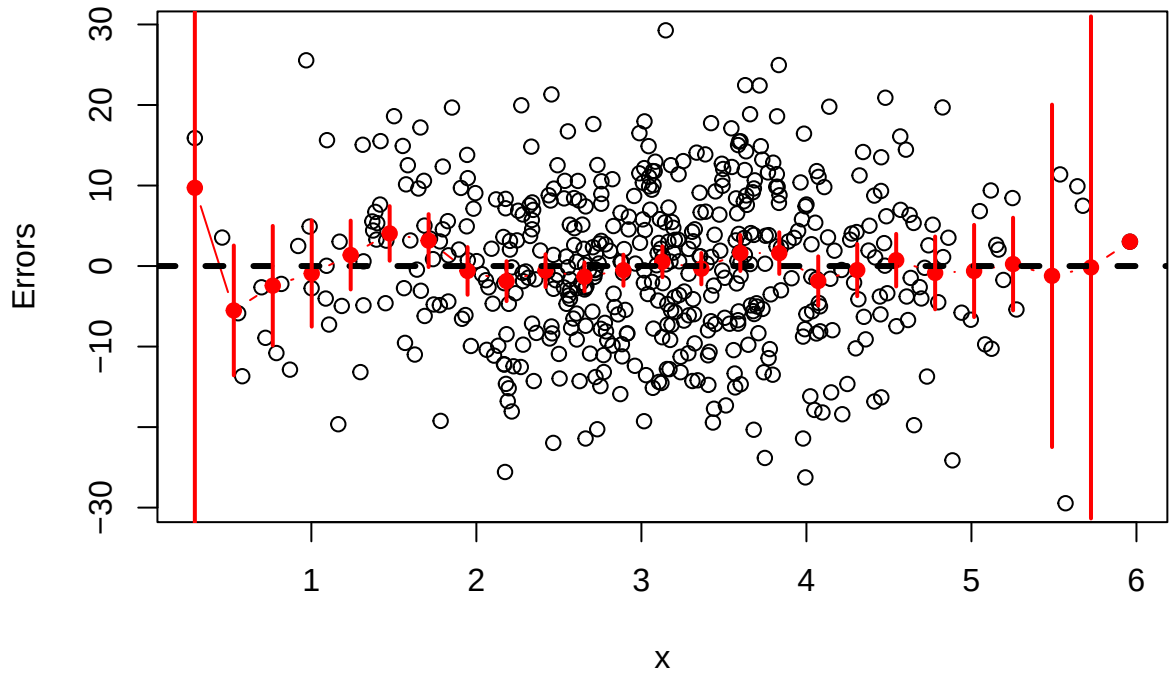


We can now check whether this satisfies the zero conditional mean assumption. For this purpose, the mean residuals are calculated for different x values:

```
y.range <- range(data.1$error)
x.range <- range(data.1$X1)

plot(data.1$error ~ data.1$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 <- data.1$error[(data.1$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)
}
```



The red dots are the mean residuals conditional to different x values. For each dots, their 95% confidence intervals are built since the data here is a random sample (the red vertical lines). The confidence intervals include the zero (the horizontal dotted lines), which indicates that we cannot reject the null hypothesis that the conditional mean is zero.

For very small and large x values, we tend to have larger deviations from zero, however, with large confidence intervals. These estimates are uncertain due to the small sample size in these areas.

The estimates of regression coefficients based on the first generated dataset

We estimate the regression line based on the first generated dataset:

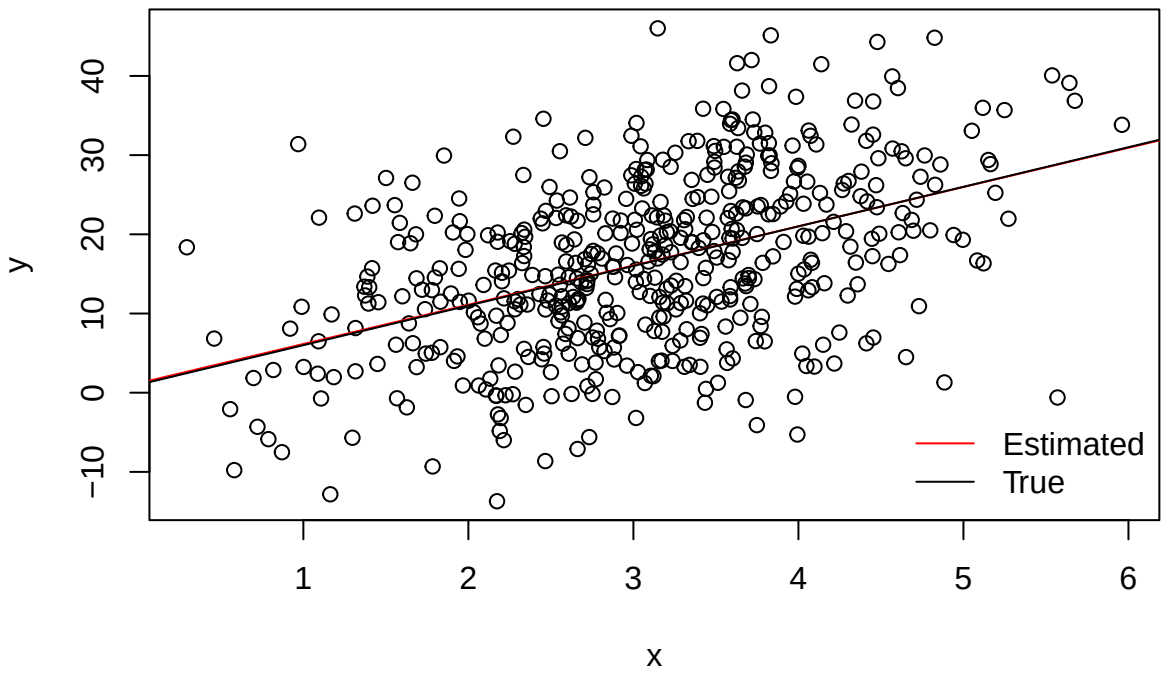
```
lm.out <- lm(y ~ X1,data=data.1)
summary(lm.out)
```

```
##
## Call:
## lm(formula = y ~ X1, data = data.1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.4022  -6.6466   0.3448   6.8984  29.2164
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.1936     1.4136   0.844    0.399
## X1            4.9558     0.4367  11.349   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.791 on 498 degrees of freedom
## Multiple R-squared:  0.2055, Adjusted R-squared:  0.2039
## F-statistic: 128.8 on 1 and 498 DF,  p-value: < 2.2e-16
```

We plot the estimated regression 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=GM.samples$para)

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



The red line is the estimated regression line, while the black line is the true regression line. Both lines are similar, but slightly different due to random sampling.

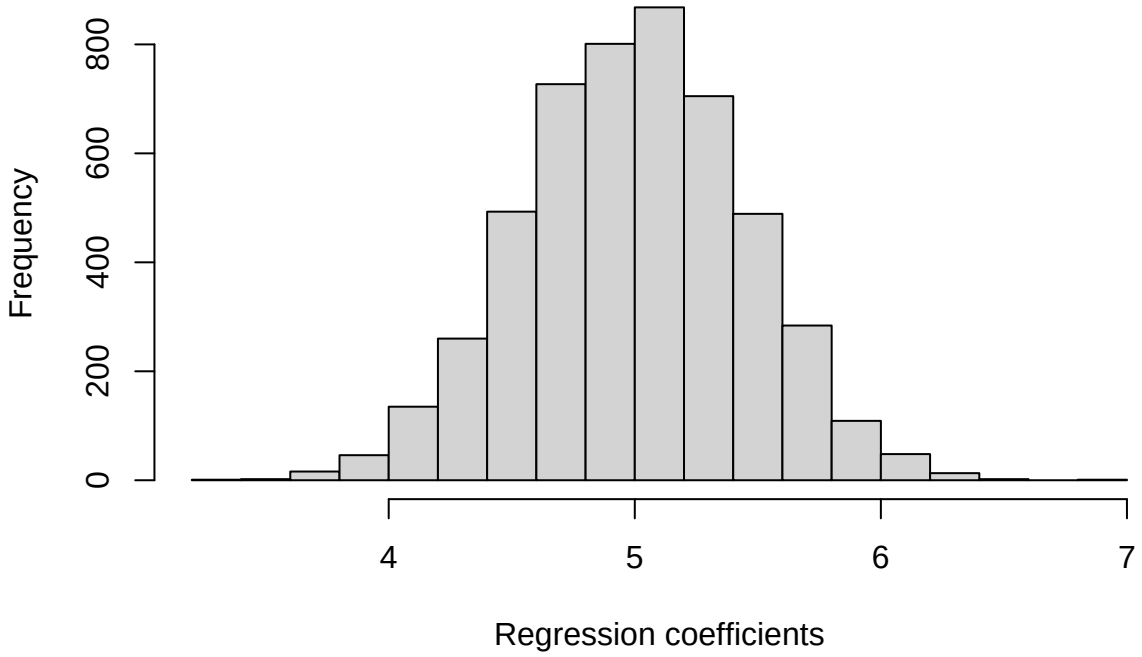
Distribution of the estimates of regression coefficients

We estimate the simple regression model by using each of 5000 datasets:

```
all.coef <- 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,data=  this.data)

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

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

```
hist(all.coef[,"b1"],main="",xlab="Regression coefficients")
```



The mean value of this distribution is 4.998. This is almost identical with the true parameter value. And if we increase the number of generated datasets, we will obtain the identical value with the truth, which means unbiasedness.

Uncerntaity of the estimate

As seen above, the estimates of regression coefficients have uncertainty, which can be measured by variance. The above distribution has variance of 0.202.

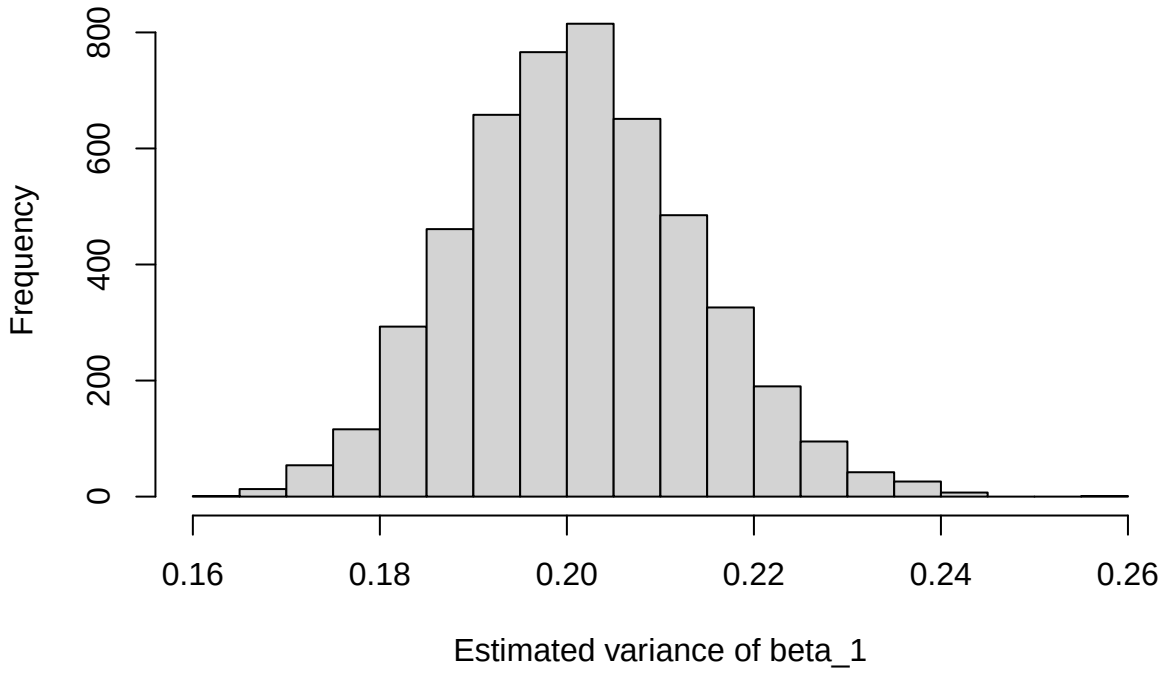
This value cannot be obtained from a single dataset, but can be estimated:

$$Var(\hat{\beta}_1) = \frac{\sigma^2}{SST_x} = \frac{\sigma^2}{\sum_i (x_i - \bar{x})^2}$$

.

```
all.var.estimate <- matrix(NA,nrow=num.datasets,ncol=3)

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

    this.SST <- sum((this.data$X1 - mean(this.data$X1))^2)

    all.var.estimate[i,1] <- true.err.var
    all.var.estimate[i,2] <- this.SST
    all.var.estimate[i,3] <- true.err.var/this.SST
}
colnames(all.var.estimate) <- c("sigma","SST","var.beta")
```

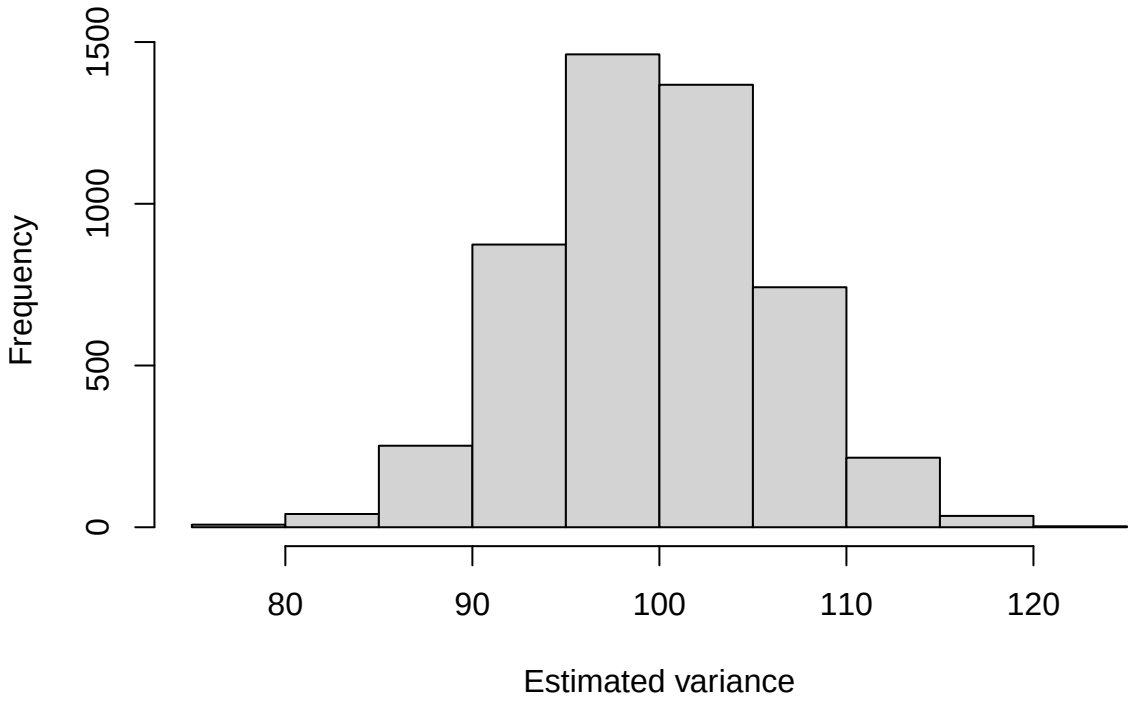
```r
hist(all.var.estimate[,"var.beta"],main="",xlab="Estimated variance of beta_1")
```



This distribution's expected value is 0.201, which corresponds to the above variance.

The problem of this approach is that we do not know the true value of error variance in most situations. Therefore, we have to estimate the error variance. For this purpose, we can utilize the residuals, which come from the observed data and the estimated regression line. However, their variance is a biased estimator of the error variance.

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

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

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

    all.error.estimate[i] <- naive.var(lm.out$residuals)
}


hist(all.error.estimate,main="",xlab="Estimated variance")
```



The above distribution's expected value is 99.6375. This is similar to the true error variance 100, but it will not converge to the true value even though we increase the number of simulated datasets.

Fortunately, we have the unbiased estimator for the error variance:

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_i \hat{u}_i^2$$

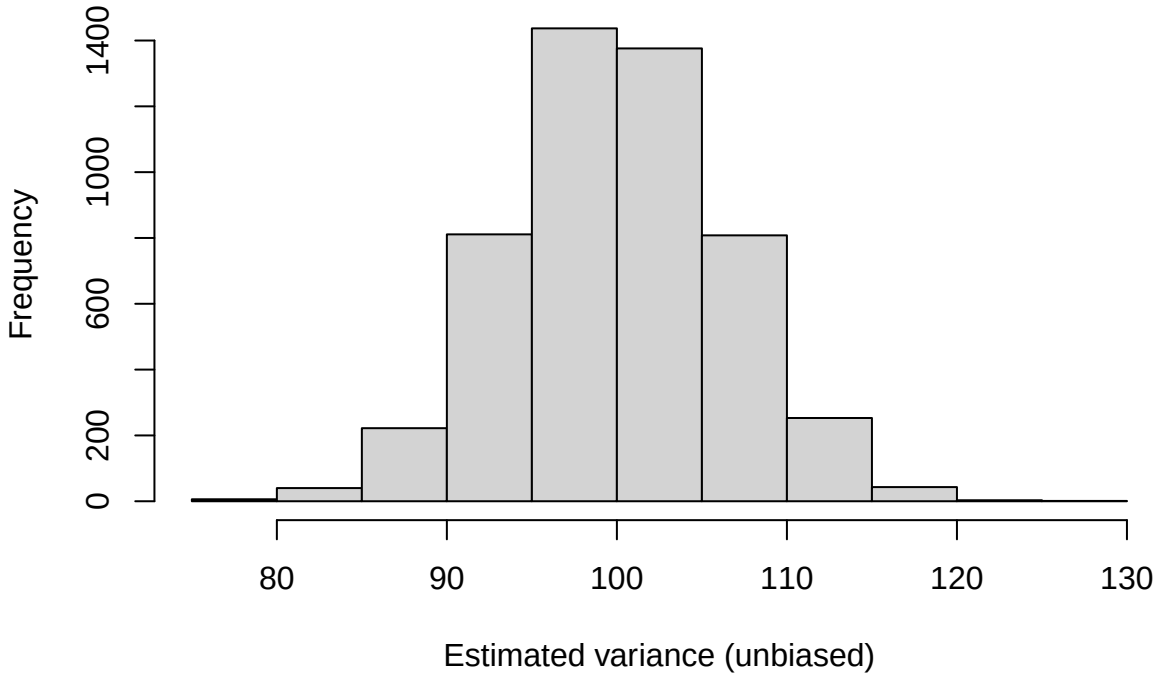
We divide by n-2 instead of n since the residuals have only n-2 degrees of freedom. In other words, if n-2 residuals are determined, the last 2 have to be automatically determined due to the restrictions in OLS:

$$\sum_i \hat{u}_i = 0$$

and

$$\sum_i x_i \hat{u}_i = 0$$

```
all.error.estimate.unbiased <- all.error.estimate*sample.size/(sample.size-2)

hist(all.error.estimate.unbiased,main="",xlab="Estimated variance (unbiased)")
```



This distribution's expected value is 100.0376. Obviously, it is closer to the true error variance.

Homoskedasticity and heteroskedasticity

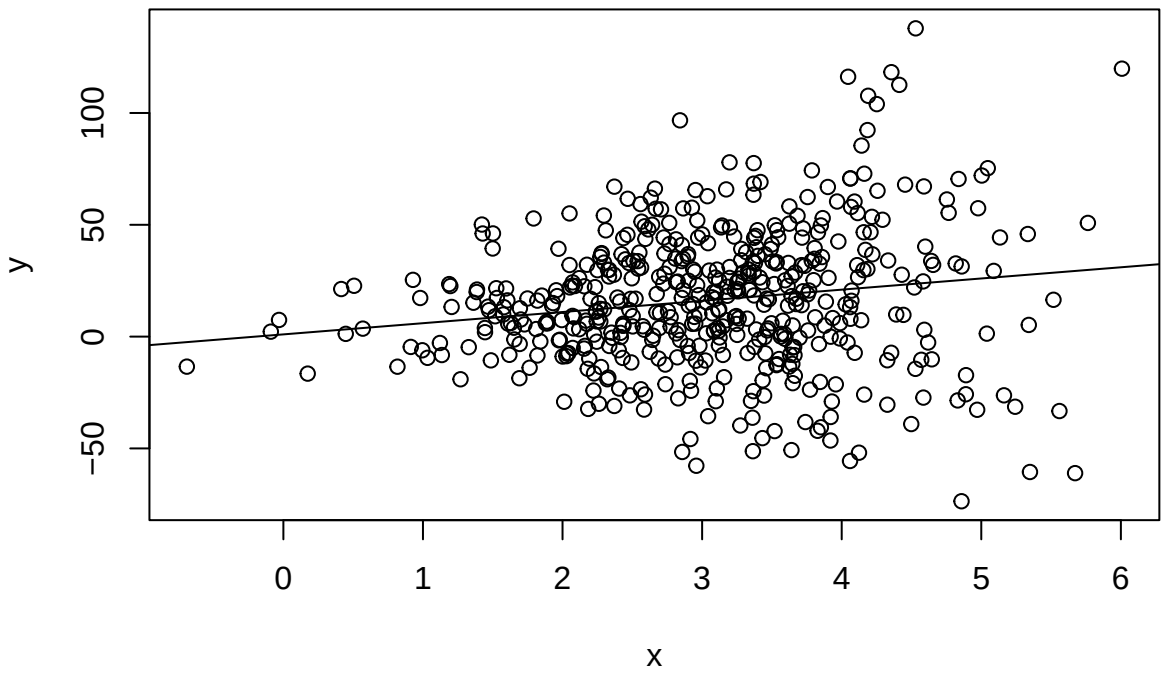
It has to be noted that the variance of OLS estimates above works only under all five GM assumptions, while OLS is garanteed to be the unbiased estimator under the first four GM assumptions (i.e. without homoskedasticity).

By using the data.generation function, we can also generate multiple datasets without the homoscedasticity assumption.

```
het.samples <- data.generation(sample.size=sample.size,
                               n.sim=num.datasets,
                               n.iv=1,
                               x.mu=3,
                               x.Sigma=as.matrix(1,nrow=1),
                               para=c(true.intercept,true.slope),
                               err.dist = "normal",
                               err.disp = true.err.var,
                               het=TRUE,
                               het.delta = c(0.5,0.5))
```

The first generated dataset looks as follows:

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

plot(data.1$y ~ data.1$X1,ylab="y",xlab="x")
abline(coef=het.samples$para)
```



It is clearly to see that the errors are larger for larger x values, and vice versa.