Simulation studies for simple linear regression model

In Section 9, we learned how to use the least squares method to estimate a simple linear regression model. One important property of the least square estimators is the “unbiasedness”, i.e.,  and . This means that if many, many samples were taken from a population and the sample regression model was calculated each time, the average of the  and  values would be approximately β1 and β0. We also learned the variance formula of the least square estimators:

 & 

In today’s lab, we will verify the unbiasedness and variance formula of  and  through simulation evidence.

Remember that E(Y) = β0 + β1x is the mean value of Y for a specific value of x. Let x be a fixed sample of size 30 drawn from 1 to 100. Then for each x value, we simulate Y values from a normal distribution with E(Y) = β0 + β1x and Var(Y) = , where β0 = 10, β1 = 2, and  = 20. We intentionally chose a large value for  so that we can see clearly the difference between the sample regression model and the population regression model. For other parameter values, the results should be similar in nature. The scatter plot of one simulated data set along with the sample regression model and the population regression model is:

> set.seed(1234)

> x <- sample(x = 100, size = 30, replace = TRUE)

> beta0 <- 10

> beta1 <- 2

> sigmae <- 20

> E.y <- beta0 + beta1 \* x

> y <- rnorm(n=30, mean=E.y, sd=sigmae)

> plot(x = x, y = y, main = "Population and sample regression model",

xlim = range(x), ylim = c(-20, 210),

pch = 1, cex = 1.0, panel.first = grid(col = "gray", lty = "dotted"))

> mod.fit <- lm(formula = y ~ x)

> mod.fit

Call:

lm(formula = y ~ x)

Coefficients:

(Intercept) x

-1.442 1.994

> curve(expr = mod.fit$coefficients[1] + mod.fit$coefficients[2]\*x,

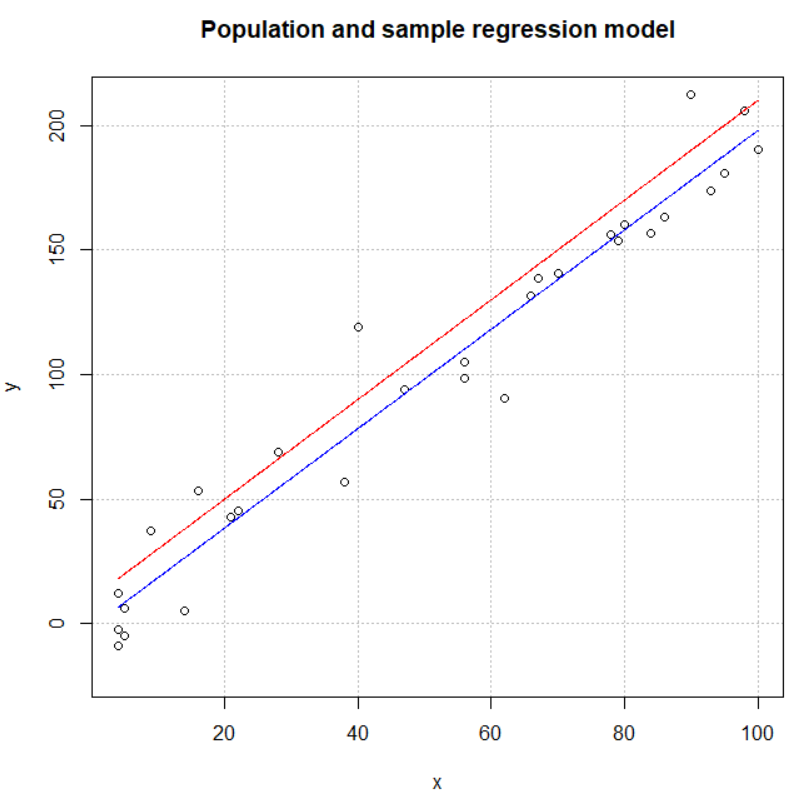
from = min(x), to = max(x), col = "blue", add = TRUE, n = 1000,

lwd = 1)

> curve(expr = beta0 + beta1 \* x,

from = min(x), to = max(x), col = "red", add = TRUE, n = 1000,

lwd = 1)



where the blue line corresponds to the sample regression model based on  and , and the red line corresponds to the population regression model. Note that in the rnorm() function, the mean argument is set as a vector E.y. Each y value is then simulated using each element of E.y as its mean.

The plot is similar to the population and sample plots in the course notes. One can think of the y values here as samples from the normal distribution with mean E(Y) and variance  Because there are “unsampled” values from the population, the sample regression model is different from the population regression model (the two lines do not overlap).

Next, using the same parameter values and x values, we simulate 10 data sets and estimate a sample regression model for each data set. Then we plot all 10 sample regression models as well as the population regression model in the same graph.

> sample.model <- function(y) {

mod.fit <- lm(formula = y ~ x)

beta0 <- mod.fit$coefficients[1]

beta1 <- mod.fit$coefficients[2]

curve(expr = beta0 + beta1 \* x,

from = min(x), to = max(x), col = "blue", add = TRUE, n = 1000,

lwd = 1)

c(beta0, beta1)

}

> set.seed(2574)

> y <- matrix(rnorm(n = 30\*10, mean = E.y, sd = sigmae), nrow = 10, byrow = TRUE)

> plot(x = x, y = y[1,], type = "n", ylab="y", main = "Population and sample

regression model", col = "red", xlim = range(x), ylim = c(0, 210),

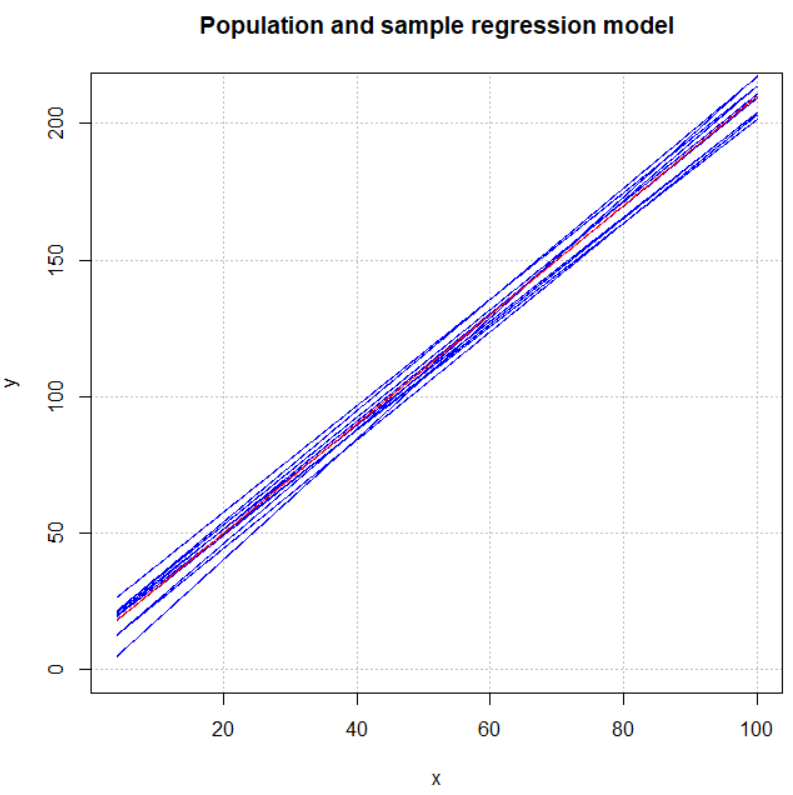
panel.first = grid(col = "gray", lty = "dotted"))

> est <- apply(X = y, MARGIN = 1, FUN = sample.model)

> curve(expr = beta0 + beta1 \* x,

from = min(x), to = max(x), col = "red", add = TRUE, n = 1000,

lwd = 1)



We again use the matrix() function in simulating the data. Now each row of y contains a sample of size n = 30. We wrote the function sample.model() to calculate  and  for each sample and to add the regression model to the existing plot. Note in the plot() function, the type = "n" argument tells R to only construct the x and y axis without plotting the data. We used this option because we don’t want to display the original data points in this graph.

In the above plot, we can see that all the blue lines (sample regression model) tend to center around the red line (population regression model). Moreover, the object est now contains all the estimates from the 10 data sets, so we can use the apply() function to find the average of the two estimates.

> est

[,1] [,2] [,3] [,4] [,5]

(Intercept) 10.476578 13.250562 4.314158 -3.952247 4.74179

x 1.934872 1.905524 2.093775 2.213230 1.98405

[,6] [,7] [,8] [,9]

(Intercept) 11.92821 18.692647 13.594799 12.049811

x 1.89554 1.949663 1.971276 1.972253

[,10]

(Intercept) 13.354199

x 2.037358

> apply(X = est, MARGIN = 1, FUN = mean)

(Intercept) x

9.845051 1.995754

The average of  and  are relatively close to the true values β0 = 10 and β1 = 2, which is consistent with the plot.

We repeated the same process for 100 data sets. The results are:

> set.seed(2574)

> y <- matrix(rnorm(n = 30\*100, mean = E.y, sd = sigmae), nrow = 100, byrow = TRUE)

> plot(x = x, y = y[1,], type = "n", ylab="y", main = "Population and sample

regression model", col = "red", xlim = range(x), ylim = c(0, 210),

panel.first = grid(col = "gray", lty = "dotted"))

> est <- apply(X = y, MARGIN = 1, FUN = sample.model)

> curve(expr = beta0 + beta1 \* x,

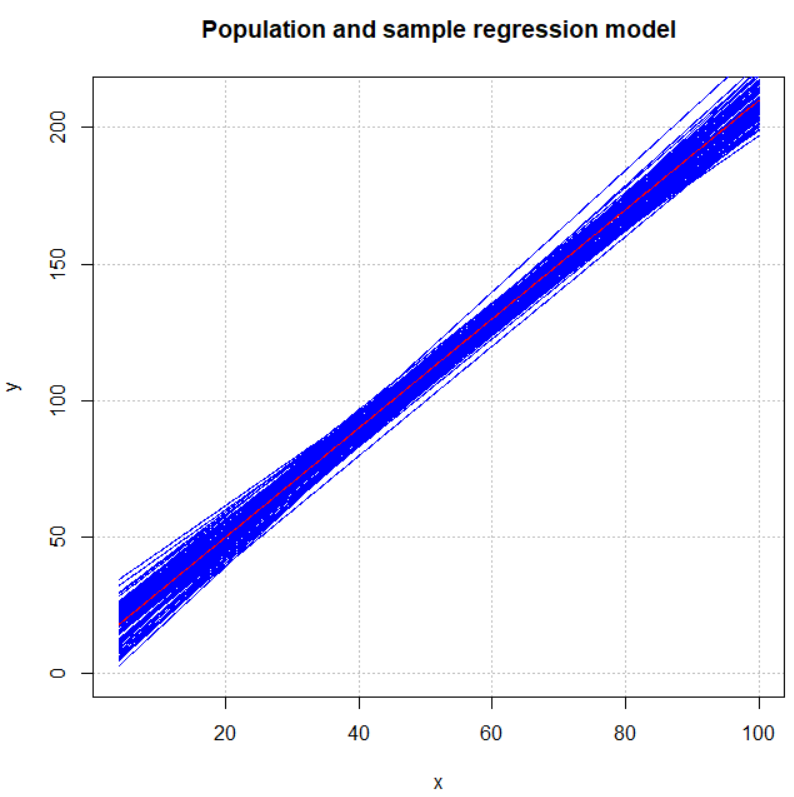
from = min(x), to = max(x), col = "red", add = TRUE, n = 1000,

lwd = 1)

> apply(X=est, MARGIN=1, FUN=mean)

(Intercept) x

10.002196 1.990451



The sample regression models center around the population regression model. With increased number of samples, the average of  and  are much close to the true values β0 and β1. Our simulation evidence supports the unbiasedness property of the least square estimators.

In the above experiments, we can visualize the variability of  and  by plotting the estimated regression models from all the simulated data sets. More precisely, the variability can be measured by the sample variance of all  and :

> apply(X = est, MARGIN = 1, FUN = var)

(Intercept) x

45.42252487 0.01238486

In real applications, we often will not be able to obtain more than one sample from the population, and we don’t know what the true parameters are. How can we still assess the variability of  and ? We can use the following two formulas!

 & 

If we substitute the x values and  into the above formulas, we get  = 0.012 and  = 44.048, which close to the sample variance estimates.

> sigmae^2/sum((x-mean(x))^2)

[1] 0.01201224

> sigmae^2\*(1/30+mean(x)^2/sum((x-mean(x))^2))

[1] 44.04849

Obviously, in practice we will not know what the unknown population parameter  is. Instead, we use the mean square error  in the above formulas to get:

 & 

where



For each data set, R will calculate  and  using the above formulas. In the previous example with 100 data sets, the 1st data set yields the following estimates:

> mod.fit <- lm(formula = y[1,] ~ x)

> summary(mod.fit)

Call:

lm(formula = y[1, ] ~ x)

Residuals:

Min 1Q Median 3Q Max

-42.700 -14.759 -2.706 14.217 50.775

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 10.477 7.451 1.406 0.171

x 1.935 0.123 15.726 1.99e-15 \*\*\*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 22.45 on 28 degrees of freedom

Multiple R-squared: 0.8983, Adjusted R-squared: 0.8947

F-statistic: 247.3 on 1 and 28 DF, p-value: 1.992e-15

Hence,  = 0.1232 = 0.015 and  = 7.4512 = 55.5174.

The 10th data set yields the following estimates:

> mod.fit<-lm(formula = y[10,] ~ x)

> summary(mod.fit)

Call:

lm(formula = y[10, ] ~ x)

Residuals:

Min 1Q Median 3Q Max

-43.294 -10.171 0.352 16.944 40.213

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.3542 7.0417 1.896 0.0683 .

x 2.0374 0.1163 17.520 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.45 on 28 degrees of freedom

Multiple R-squared: 0.9071, Adjusted R-squared: 0.9038

F-statistic: 273.4 on 1 and 28 DF, p-value: 5.58e-16

Hence,  = 0.11632 = 0.014 and  = 7.04172 = 49.59.

These numbers are relatively close to the population variances  = 0.017 and  = 48.467, given the scale of the estimates. Here we only looked at two data sets. Notice how  and  are different across different data sets, because each data set has its own .

Question: What do you expect will happen to  and  when the sample size of each data set increases (e.g., from 30 to 100)? Will they get closer or farther from the population variances (assuming we know what the true parameters are)?