PredictiveAssignment-2

Shreyansh Kumar Das

2026-02-02

Problem Set 2: Linear Regression

1. Problem to demonstrate that the population regression line is fixed, but least square regression line varies.

Suppose the population regression line is given by *Y = 2+ 3x*, while the data comes from the model *y = 2 + 3x*.
Step 1: For *x* in range [5,10], graph the population regression line.
Step 2: Generate x_i *(i=1,2,...,n)* from *Uniform(5,10)* and the ε_i*(i=1,2,...,n)* from *N(0,1)*. Hence, compute the $y_1, y_2, .., y_n$.
Step 3: Ont he basis if the data (x_i, y_i)*(i=1,2,...,n*) generated in Step 2, report the least squares regression line.
Step 4: Repeat steps 2-3 fibve times. Graph the 5 least squared regression lines over the population regression line obtained in Step 1. Interpret the findings. Take *n=50*. Set the seed as 123.

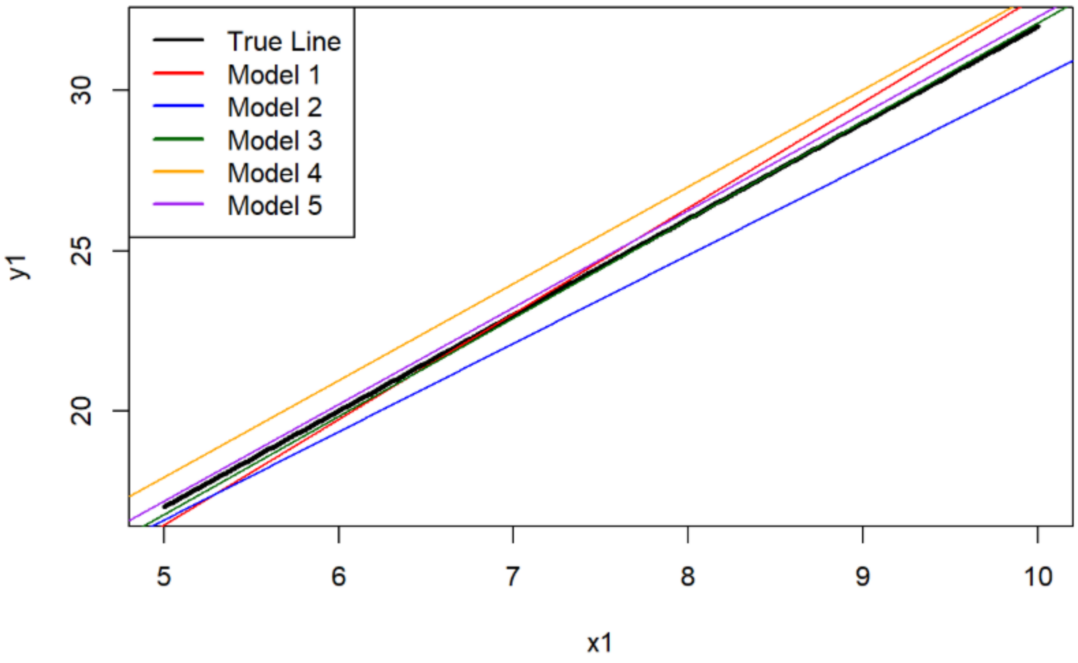
```r
set.seed(123)
x1=seq(5,10,length.out=200)
y1=2+3*x1
plot(x1,y1,type='l',col="black",lwd=3)

sim_results <- data.frame(
  iteration = 1:5,
  intercept = numeric(5),
  slope = numeric(5)
)

cols <- c("red", "blue", "darkgreen", "orange", "purple")

for (i in 1:5){
  x = runif(50, 5, 10)
  eps = rnorm(50, 0, 4)
  y = 2 + 3*x + eps
  mod = lm(y ~ x)
  abline(mod, col = cols[i])
  sim_results$intercept[i] =coef(mod)[1]
  sim_results$slope[i]= coef(mod)[2]
}
```

```
legend("topleft",
       legend = c("True Line", paste("Model", 1:5)),
       col = c("black", cols),
       lty = 1,
       lwd = 2)
```



```
sim_results

##   iteration   intercept    slope
## 1         1 -0.09638929 3.305396
## 2         2  2.79218839 2.761042
## 3         3  1.39299737 3.073267
## 4         4  2.82308856 3.023608
## 5         5  2.03250638 3.028097
```

Interpretation: The plot shows that although the true relationship is $y = 2 + 3x$, the estimated regression lines vary slightly across repeated samples due to random error, demonstrating the sampling variability of the least squares estimators of the slope and intercept.

2. Problem to demonstrate that the population regression line is fixed, but least square regression line varies.

Step 1: Generate x_i from *Uniform(5,10)* and mean centre the values. Generate ε_i form the *N(0,1)*. Calculate $y_i = 2 + 3x + \varepsilon_i$, i=1,2,...,n. Take n=50 and seed=123.

Step 2: Now imagine that you only have the data on (x_i, y_i), *i=1,2,...,n*, without knowing the mechanism that was used to generate the data step 1. Assuming a linear regression of the type $y_i = \beta_0 + \beta x_i + \varepsilon_i$ and based on these data (x_i, y_i), *i=1,2,...,n*, obtain the least squares estimates of β_0 and β.
Step 3: take a large number of grid values of (β_0, β) that also include the least squares estimates obtained from step 2. Compute the RSS for each parametric choice of (β_0, β), where $RSS = (y_1 - \beta_0 - \beta x_1)^2 + (y_2 - \beta_0 - \beta x_2)^2 + \ldots + (y_n - \beta_0 - \beta x_n)^2$.Find out for which combination of (β_0, β), RSS is minimum.

```
set.seed(123)

x_1 = runif(50, 5, 10)
eps = rnorm(50, 0, 1)

x_mc = x_1 - mean(x_1)
y = 2 + 3*x_mc + eps

mod_2 <- lm(y ~ x_mc)
b0_hat <- coef(mod_2)[1]
b1_hat <- coef(mod_2)[2]

b0_grid = seq(b0_hat - 0.5, b0_hat + 0.5, length.out = 51)
b1_grid = seq(b1_hat - 0.5, b1_hat + 0.5, length.out = 51)

RSS <- matrix(NA, 51, 51)

for (i in 1:51) {
  for (j in 1:51) {
    RSS[i, j] <- sum((y - b0_grid[i] - b1_grid[j] * x_mc)^2)
  }
}

which(RSS == min(RSS), arr.ind = TRUE)

##      row col
## [1,]  26  26
```

Interpretation: The RSS attains its minimum at grid index (26, 26), which corresponds to the least squares estimates $(\hat{\beta}_0, \hat{\beta})$.

3. Problem to demonstrate that least square estimators are unbiased.

Step 1: Generate $x_i (i = 1,2,3, \ldots, n)$ from *Uniform(0,1)*, $\epsilon_i (i = 1,2, \ldots, n)$ from *N(0,1)* and hence generate y using $y_i = \beta_0 + \beta x_i + \epsilon_i$. (Take $\beta_0 = 2, \beta = 3$).

Step 2: On the basis of the data $(x_i, y_i)(i = 1,2,3, \dots, n)$ generated in the Step 1, obtain the least square estimates of β_0 and β. Repeat the steps 1-2, $R = 1000$ times. In each simulation obtain $\hat{\beta}_0$ and $\hat{\beta}$. Finally, the least- square estimates will be given by the average of these estimated values. Compare these with the true β_0 and β and comment. Take n = 50 and seed = 123.

```
set.seed(123)

result = data.frame(
  beta0_hat = numeric(1000),
  beta_hat = numeric(1000)
)

b0 = 2
b1 = 3

for (i in 1:1000){
  x_sim = runif(50, 0, 1)
  eps = rnorm(50, 0, 1)
  y = b0 + b1*x_sim + eps
  mod_sim = lm(y ~ x_sim)

  result$beta0_hat[i] = coef(mod_sim)[1]
  result$beta_hat[i] = coef(mod_sim)[2]
}

avg_b0 = mean(result$beta0_hat); avg_b0

## [1] 2.013053

avg_b1 = mean(result$beta_hat); avg_b1

## [1] 2.982112

var_beta0 = var(result$beta0_hat); var_beta0

## [1] 0.07969639

var_beta1 = var(result$beta_hat); var_beta1

## [1] 0.2360544
```

Interpretation: From the simulation, the averages of $\hat{\beta}_0$ and $\hat{\beta}$ over 1000 repetitions are close are very close to the true parameters values $\beta_0 = 2$ and $\beta = 3$. This confirms that the least squares estimators are unbiased. The variances show that the estimates fluctuate across repeated samples, but the fluctuations average out as the number of simulations increases.

4. Comparing several simple linear regressions.

Attach “Boston” data from MASS library in R. Select median value of owner occupied homes, as the response and per capita crime rate, nitrogen oxides concentration, proportion of blacks and percentage of lower status of the population as predictors.
(a) Selecting the predictors one by one, run four separate linear regressions to the data. Present the output in a single table.
(b) Which model gives the best fit?
(c) Compare the coefficients of the predictors from each model and comment on the usefulness of the predictors.

```r
library(MASS)
data("Boston")

# (a) Fit 4 separate simple linear regression models
m1 = lm(medv ~ crim,  data = Boston)
m2 = lm(medv ~ nox,   data = Boston)
m3 = lm(medv ~ black, data = Boston)
m4 = lm(medv ~ lstat, data = Boston)


get_model_row = function(model, predictor_name){
  s <- summary(model)
  data.frame(
    Predictor = predictor_name,
    Intercept = coef(model)[1],
    Slope     = coef(model)[2],
    StdError  = s$coefficients[2, 2],
    t_value   = s$coefficients[2, 3],
    p_value   = s$coefficients[2, 4],
    R_squared = s$r.squared,
    Adj_R2    = s$adj.r.squared,
    stringsAsFactors = FALSE
  )
}

output_table = rbind(
  get_model_row(m1, "crim"),
  get_model_row(m2, "nox"),
  get_model_row(m3, "black"),
  get_model_row(m4, "lstat")
)
output_table
```

```
##                Predictor Intercept        Slope    StdError    t_value
## (Intercept)        crim  24.03311  -0.41519028 0.043890381  -9.459710
## (Intercept)1        nox  41.34587 -33.91605501 3.196337032 -10.610913
## (Intercept)2      black  10.55103   0.03359306 0.004230507   7.940669
## (Intercept)3      lstat  34.55384  -0.95004935 0.038733416 -24.527900
##                    p_value R_squared    Adj_R2
## (Intercept)  1.173987e-19 0.1507805 0.1490955
## (Intercept)1 7.065042e-24 0.1826030 0.1809812
## (Intercept)2 1.318113e-14 0.1111961 0.1094326
## (Intercept)3 5.081103e-88 0.5441463 0.5432418
```

```
# (b) Which model gives the best fit?
best_model = output_table[which.max(output_table$R_squared), ]
best_model
```

```
##                Predictor Intercept      Slope   StdError  t_value
p_value
## (Intercept)3      lstat  34.55384 -0.9500494 0.03873342 -24.5279 5.081103e-
88
##                R_squared    Adj_R2
## (Intercept)3 0.5441463 0.5432418
```

```
# (c) Compare coefficients and comment usefulness
output_table[, c("Predictor", "Slope", "p_value", "R_squared")]
```

```
##                Predictor        Slope      p_value R_squared
## (Intercept)        crim  -0.41519028 1.173987e-19 0.1507805
## (Intercept)1        nox -33.91605501 7.065042e-24 0.1826030
## (Intercept)2      black   0.03359306 1.318113e-14 0.1111961
## (Intercept)3      lstat  -0.95004935 5.081103e-88 0.5441463
```