

Problem Set 2

Meghna

2026-02-05

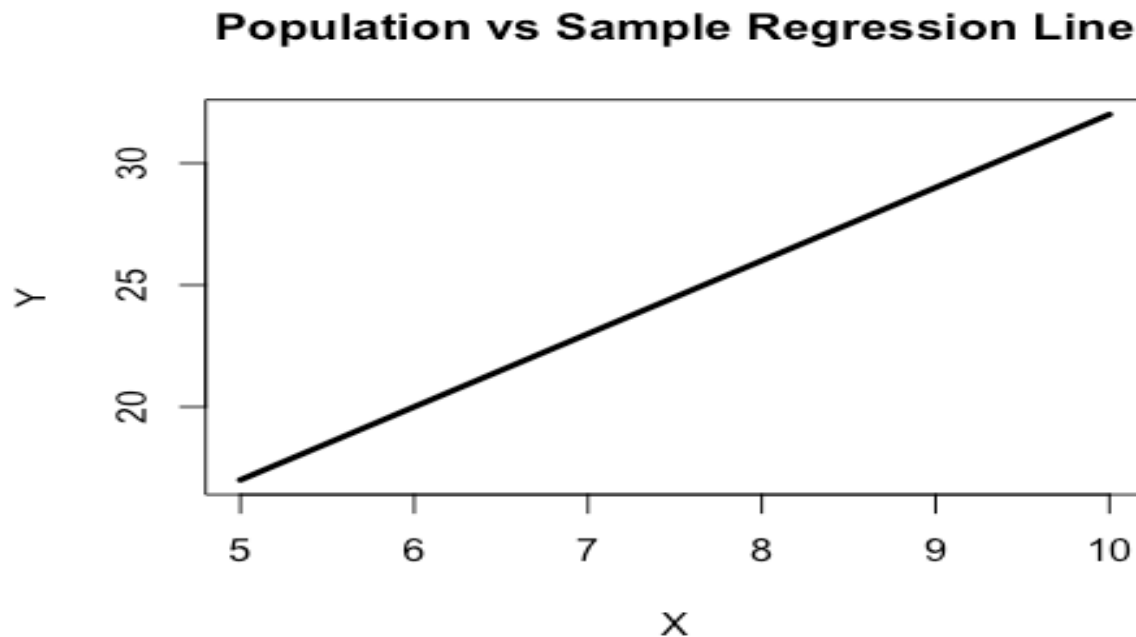
1. Problem to demonstrate that the population regression line is fixed, but least square regression line varies. Suppose the population regression line is given by $Y = 2 + 3x$, while the data comes from the model $y = 2 + 3x + \epsilon$.

```
rm(list=ls())
```

Step 1: For x in the range $[5, 10]$ graph the population regression line.

```
x_pop <- seq(5, 10, length.out = 100)
y_pop <- 2 + 3 * x_pop

plot(x_pop, y_pop, type = "l", lwd = 3, col = "black",
     xlab = "X", ylab = "Y",
     main = "Population vs Sample Regression Line")
```



Step 2: Generate $x_i (i = 1, 2, \dots, n)$ from $\text{Uniform}(5, 10)$ and $\epsilon_i (i = 1, 2, \dots, n)$ from $N(0, 42)$. Hence, compute y_1, y_2, \dots, y_n .

```
n <- 50
set.seed(123)
```

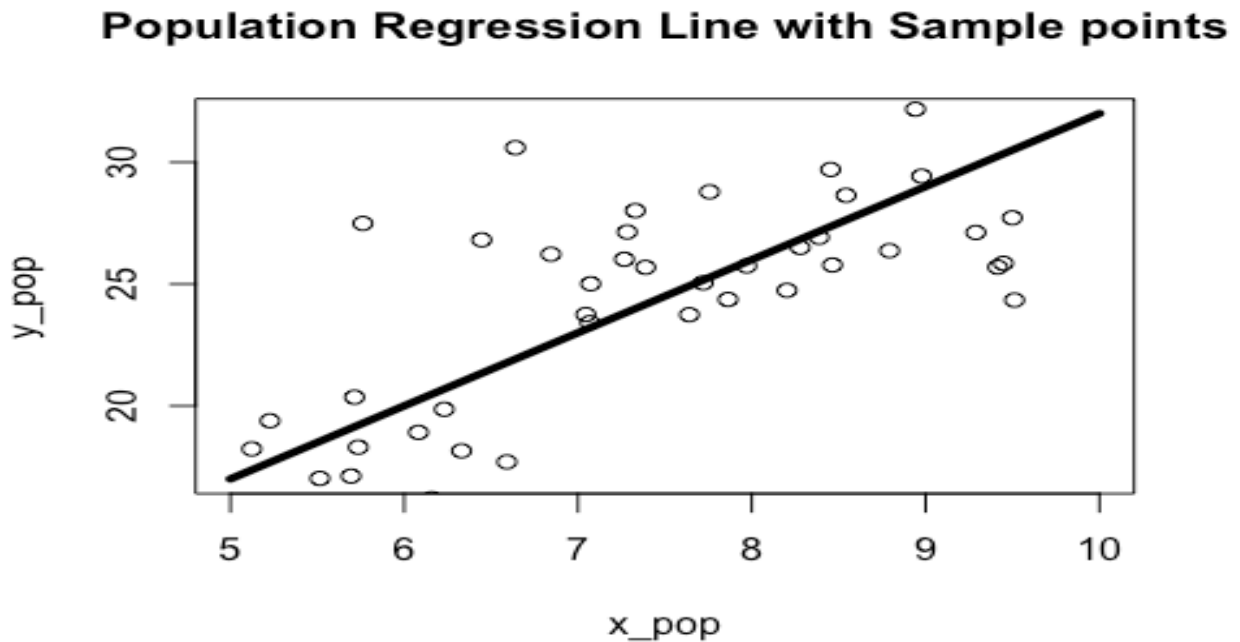
```

xi <- runif(n,5,10)
ei <- rnorm(n,0,4)
yi <- 2 + 3 * xi + ei

plot(x_pop, y_pop,type='l', lwd=4,
     main="Population Regression Line with Sample points")

points(xi,yi)

```



Step 3: On the basis of the data $(x_i, y_i) (i = 1, 2, \dots, n)$ generated in Step 2, report the least squares regression line.

```

model1 <- lm(yi~xi)

summary(model1)

##
## Call:
## lm(formula = yi ~ xi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.0231 -2.2314 -0.2627  2.1970  8.7445
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.09639    2.82610  -0.034   0.973
## xi           3.30540    0.36519   9.051 5.96e-12 ***

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.761 on 48 degrees of freedom
## Multiple R-squared:  0.6306, Adjusted R-squared:  0.6229
## F-statistic: 81.93 on 1 and 48 DF,  p-value: 5.962e-12

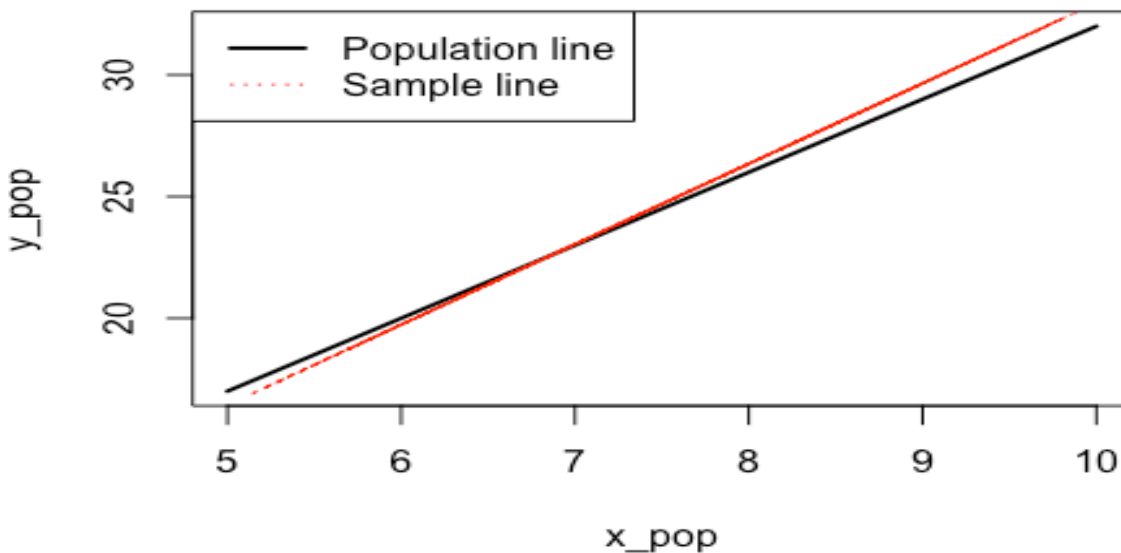
y_hat <- -0.09639+3.30540*xi

plot(x_pop, y_pop, type='l',lwd=2,
     main="Population Regression Line with Least Square Regression line ")

lines(xi,y_hat,type='l',col="red",lty="dotted")

legend("topleft",col=c("black","red"),lwd=c(2,1),legend=c("Population
line","Sample line"),
      lty=c("solid","dotted"))
```

Population Regression Line with Least Square Regression



Step 4: Repeat steps 2-3 five times. Graph the 5 least squares regression lines over the population regression line obtained in Step 1. Take $n = 50$. Set the seed as seed=123. Interpret the findings.

```
set.seed(123)
n <- 50

# Create empty dataframe to store coefficients
coef_df <- data.frame(
  Intercept = numeric(),
```

```

    Slope = numeric()
  )

plot(x_pop, y_pop, type = "l", lwd = 3, col = "black",
     xlab = "X", ylab = "Y",
     main = "Population Regression Line with Sample Regression Lines")

# Generate samples and add regression lines
for(i in 1:5){
  x <- runif(n, 5, 10)
  eps <- rnorm(n, mean = 0, sd = 4)
  y <- 2 + 3 * x + eps

  fit <- lm(y ~ x)

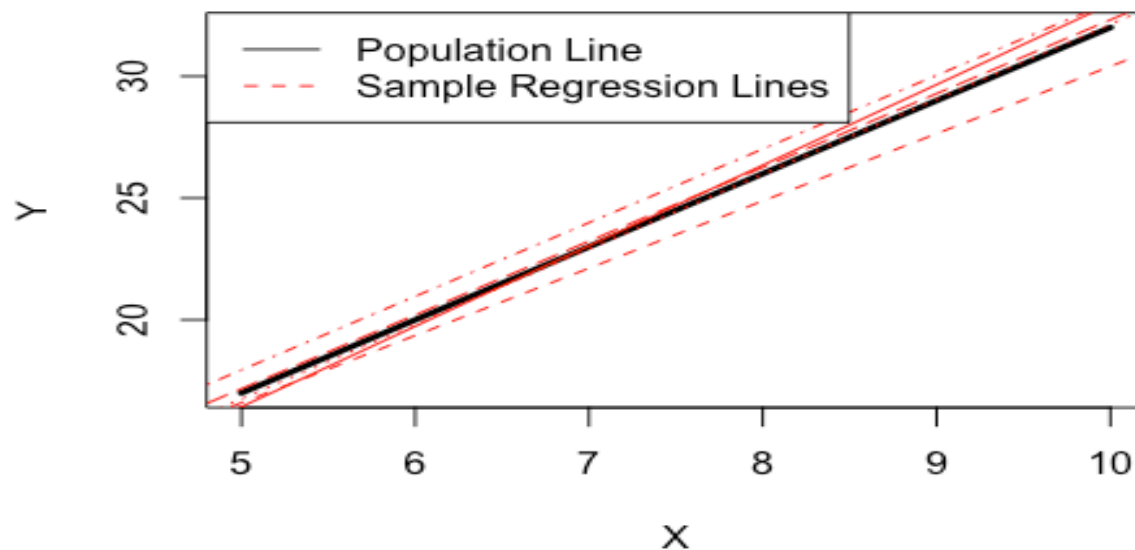
  # Store coefficients
  coef_df <- rbind(
    coef_df,
    data.frame(
      Intercept = coef(fit)[1],
      Slope = coef(fit)[2]
    )
  )

  # Add regression line
  abline(fit, col = "red", lty = i)
}

legend("topleft",
      legend = c("Population Line", "Sample Regression Lines"),
      col = c("black", "red"),
      lty = c(1, 2))

```

Population Regression Line with Sample Regression Lines



```
# Display coefficient table
coef_df

##           Intercept      Slope
## (Intercept) -0.09638929 3.305396
## (Intercept)1  2.79218839 2.761042
## (Intercept)2  1.39299737 3.073267
## (Intercept)3  2.82308856 3.023608
## (Intercept)4  2.03250638 3.028097
```

Interpretation:

The population regression line remains fixed, while the least squares regression lines vary across samples. As sample size increases, these estimated lines tend to cluster around the population line.

2. Problem to demonstrate that $\hat{\beta}_0$ and $\hat{\beta}_1$ minimises RSS

```
rm(list=ls())
```

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

```
set.seed(123)
n <- 50
x <- runif(n, 5, 10)
x <- x - mean(x)
eps <- rnorm(n)
y <- 2 + 3 * x + eps
```

Step 2: Now imagine that you only have the data on $(x_i, y_i), i = 1, 2, \dots, n$, without knowing the mechanism that was used to generate the data in step 1. Assuming a linear regression of the type $y_i = \beta_0 + \beta x_i + \epsilon_i$, and based on these data $(x_i, y_i), i = 1, 2, \dots, n$, obtain the least squares estimates of β_0 and β .

```
fit <- lm(y ~ x)
coef(fit)

## (Intercept)          x
##    2.056189    3.076349
```

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 $RSS = (y_1 - \beta_0 - \beta x_1)^2 + (y_2 - \beta_0 - \beta x_2)^2 + \dots + (y_n - \beta_0 - \beta x_n)^2$. Find out for which combination of (β_0, β) , RSS is minimum.

```
b0_grid <- seq(0, 4, length = 50)
b1_grid <- seq(1, 5, length = 50)
RSS <- matrix(NA, nrow = 50, ncol = 50)

for(i in 1:50){
  for(j in 1:50){
    RSS[i, j] <- sum((y - b0_grid[i] - b1_grid[j] * x)^2)
  }
}

min(RSS)

## [1] 42.59126
```

Interpretation:

The Least Square Estimates minimizes the RSS.

3. Problem to demonstrate that least square estimators are unbiased

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

```
n <- 50
x <- runif(n, 0, 1)
eps <- rnorm(n, 0, 1)
y <- 2 + 3 * x + eps
```

Step 2: On the basis of the data $(x_i, y_i) (i = 1, 2, \dots, n)$ generated in Step 1, obtain the least square estimates of β_0 and β . Repeat 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)
R <- 1000
b0_hat <- numeric(R)
b1_hat <- numeric(R)
n <- 50

for(i in 1:R){
  x <- runif(n, 0, 1)
  eps <- rnorm(n, 0, 1)
  y <- 2 + 3 * x + eps
  fit <- lm(y ~ x)
  b0_hat[i] <- coef(fit)[1]
  b1_hat[i] <- coef(fit)[2]
}

mean(b0_hat)

## [1] 2.013053

mean(b1_hat)

## [1] 2.982112

```

Interpretation:

The average of the estimated coefficients across simulations is very close to the true values, demonstrating that least squares estimators are unbiased.

- 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.

```

library(MASS)
data(Boston)

```

- (a) Selecting the predictors one by one, run four separate linear regressions to the data. Present the output in a single table.**

```

fit1 <- lm(medv ~ crim, data = Boston)
fit2 <- lm(medv ~ nox, data = Boston)
fit3 <- lm(medv ~ black, data = Boston)
fit4 <- lm(medv ~ lstat, data = Boston)

summary(fit1)

##
## Call:
## lm(formula = medv ~ crim, data = Boston)
##
## Residuals:

```

```
##      Min      1Q  Median      3Q      Max
## -16.957  -5.449  -2.007   2.512  29.800
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.03311    0.40914   58.74  <2e-16 ***
## crim        -0.41519    0.04389   -9.46  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.484 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

`summary(fit2)`

```
##
## Call:
## lm(formula = medv ~ nox, data = Boston)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -13.691  -5.121  -2.161   2.959  31.310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   41.346     1.811   22.83  <2e-16 ***
## nox          -33.916     3.196  -10.61  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.323 on 504 degrees of freedom
## Multiple R-squared:  0.1826, Adjusted R-squared:  0.181
## F-statistic: 112.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

`summary(fit3)`

```
##
## Call:
## lm(formula = medv ~ black, data = Boston)
##
## Residuals:
##      Min      1Q  Median      3Q      Max
## -18.884  -4.862  -1.684   2.932  27.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.551034   1.557463   6.775 3.49e-11 ***
## black        0.033593   0.004231   7.941 1.32e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.679 on 504 degrees of freedom
## Multiple R-squared:  0.1112, Adjusted R-squared:  0.1094
## F-statistic: 63.05 on 1 and 504 DF,  p-value: 1.318e-14

summary(fit4)

##
## Call:
## lm(formula = medv ~ lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.168  -3.990  -1.318   2.034   24.500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41  <2e-16 ***
## lstat       -0.95005    0.03873  -24.53  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

(b) Which model gives the best fit?

Among all models, the regression using *lstat* as predictor provides the highest R-squared value, indicating the best fit.

(c) Compare the coefficients of the predictors from each model and comment on the usefulness of the predictors.

- ***crim* and *nox* have negative effects on house prices.**
- ***black* shows a weak relationship with *medv*.**
- ***lstat* is the strongest predictor, showing a strong negative association with median house value.**