# Solutions to problem set2 (predictive analysis)

Abhilasha Das

2026-02-04

## Problem 1

Suppose the population regression line is given by $Y = 2 + 3x$, while the data comes from the model $y = 2 + 3x + \varepsilon$.

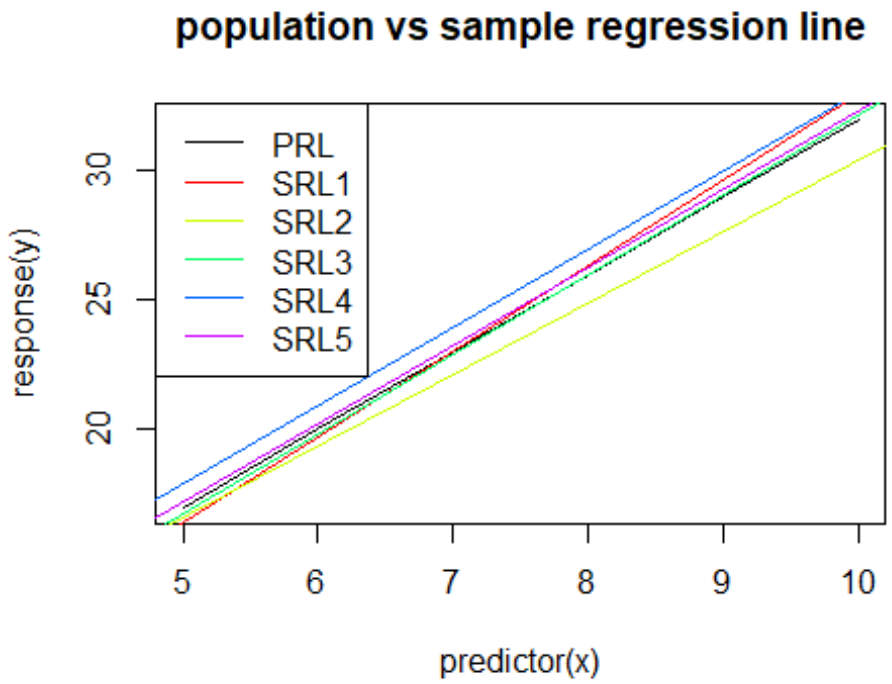
Step 1: For x in the range [5,10] graph the population regression line.

Step 2: Generate $x_i(i = 1,2,..,n)$ from Uniform(5, 10) and $\varepsilon_i(i = 1,2,..,n)$ from $N(0, 4^2)$. Hence, compute $y_1, y_2, .., y_n$.

Step 3: On the basis of 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 five times. Graph the 5 least squares regression lines over the population regression line obtained in Step 1. Interpret the findings. Take n = 50. Set the seed as seed=123.

```
rm(list=ls())
n=50;set.seed(123)
x=5:10;y=2+3*x
plot(x,y,type='l',
     xlab='predictor(x)',
     ylab='response(y)',
     main='population vs sample regression line')
for (i in 1:5){
  xi=runif(n,5,10);ei=rnorm(n,0,sd=4)
  yi=2+3*xi+ei
  abline(lm(yi~xi),col=rainbow(5)[i])
}
legend('topleft',
       legend=c('PRL','SRL1','SRL2','SRL3','SRL4','SRL5'),
       col=c('black',rainbow(5)),lty=1)
```




## Problem 2:

Demonstrating that $\hat{\beta}_0$ and $\hat{\beta}$ Minimize RSS. The objective of this exercise is to numerically demonstrate that the Ordinary Least Squares (OLS) estimates, $\hat{\beta}_0$ and $\hat{\beta}$, are the specific values that minimize the Residual Sum of Squares (RSS).

### Step 1: Data Generation

1. Generate $x_i$ from a Uniform(5,10) distribution and **mean-center** the values.
2. Generate error terms $\varepsilon_i$ from a normal distribution $N(0,1)$.
3. Calculate the dependent variable:

$$y_i = 2 + 3x_i + \varepsilon_i, \quad i = 1,2, \dots, n$$

4. Set the sample size $n = 50$ and use `set.seed(123)` for reproducibility.

### Step 2: OLS Estimation

Assume you only have access to the observed data points $(x_i, y_i)$ for $i = 1, \dots, n$, without knowing the underlying population parameters used in Step 1.

Fit a linear regression model:

$$y_i = \beta_0 + \beta x_i + \varepsilon_i$$

Based on this data, obtain the **least squares estimates** $\hat{\beta}_0$ and $\hat{\beta}$.

### Step 3: Grid Search & RSS Minimization

1. Create a large grid of potential values for $(\beta_0, \beta)$, ensuring the grid includes the OLS estimates obtained in Step 2.
2. For each parametric choice $(\beta_0, \beta)$ in the grid, compute the **Residual Sum of Squares (RSS)**:

$$RSS = \sum_{i=1}^{n} (y_i - \beta_0 - \beta x_i)^2$$

3. Identify the specific combination of $(\beta_0, \beta)$ for which the RSS is at its minimum.

```
rm(list=ls())
n=50;set.seed(123)
xi=runif(n,5,10);xi=xi-mean(xi);ei=rnorm(n)
yi=2+3*xi+ei
cof=coef(lm(yi~xi));cof

## (Intercept)          xi
##    2.056189    3.076349

b0=c(seq(0,5,0.01),cof[1]);b1=c(seq(0,5,0.01),cof[2])
rss=matrix(0,nrow=length(b0),ncol=length(b1))
for (i in 1:length(b0)){
    for (j in 1:length(b1)){
        new_yi=b0[i]+b1[j]*xi
        rss[i,j]=sum((yi-new_yi)^2)
    }
}
rss[which(rss==min(rss))]   # minimum RSS value

## [1] 42.4455

which(rss==min(rss),arr.ind=T)   # last pair of values which is the least
square estimates

##      row col
## [1,] 502 502
```

## Problem 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 $\beta_0 = 2, \beta = 3$).

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 $\beta_0$ and $\beta$. 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 $\beta_0$ and $\beta$ and comment.Take $n = 50$ and set.seed(123).

```
rm(list=ls())
n=50;set.seed(123)
b0=2;b1=3  # true value of least square estimators

R=1000;b0_hat=b1_hat=array(0)
for (i in 1:R){
    xi=runif(n,5,10);ei=rnorm(n)
    yi=b0+b1*xi+ei
    model=lm(yi~xi)
    b0_hat[i]=coef(model)[1]
    b1_hat[i]=coef(model)[2]
}
mean(b0_hat);mean(b1_hat)
```

```
## [1] 2.030941
```

```
## [1] 2.996422
```

## Problem 4:

Comparing several simple linear regressions ### Attach the "Boston" data from the MASS library in R. Select the median value of owner-occupied homes (`medv`) as the response, and the following as predictors:

`Per capita crime rate (crim)`

`Nitrogen oxides concentration (nox)`

`Proportion of blacks (black)`

`Percentage of lower status of the population (lstat)`

*(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.*

```
library(MASS)
library(stargazer)
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary
Statistics Tables.
```

```
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer

data("Boston")

# four separate linear regressions
model1 = lm(medv ~ crim, data = Boston)
model2 = lm(medv ~ nox, data = Boston)
model3 = lm(medv ~ black, data = Boston)
model4 = lm(medv ~ lstat, data = Boston)

stargazer(model1, model2, model3, model4,
          type = "text",
          title = "Comparison of Simple Linear Regressions (Response: medv)",
          column.labels = c("Crime", "NOx", "Black", "Lstat"),
          header = FALSE)

##
## Comparison of Simple Linear Regressions (Response: medv)
## ================================================================================
##                                          Dependent variable:
##                                 ---------------------------------------
##                                                  medv
##                                   Crime      NOx       Black     Lstat
##                                     (1)       (2)        (3)       (4)
## --------------------------------------------------------------------------------
## crim                              -0.415***
##                                    (0.044)
##
## nox                                         -33.916***
##                                              (3.196)
##
## black                                                  0.034***
##                                                         (0.004)
##
## lstat                                                            -0.950***
##                                                                   (0.039)
##
## Constant                          24.033*** 41.346***  10.551*** 34.554***
##                                    (0.409)   (1.811)    (1.557)   (0.563)
##
## --------------------------------------------------------------------------------
## Observations                         506       506        506       506
## R2                                  0.151     0.183      0.111     0.544
## Adjusted R2                         0.149     0.181      0.109     0.543
## Residual Std. Error (df = 504)      8.484     8.323      8.679     6.216
## F Statistic (df = 1; 504)        89.486*** 112.591*** 63.054*** 601.618***
## ================================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

```r
# Adjusted R-squared for comparison
results = data.frame(
  Predictor = c("crim", "nox", "black", "lstat"),
  Adj_R2 = c(summary(model1)$adj.r.squared,
             summary(model2)$adj.r.squared,
             summary(model3)$adj.r.squared,
             summary(model4)$adj.r.squared)
)

print(results[order(-results$Adj_R2), ])
```

```
##   Predictor    Adj_R2
## 4     lstat 0.5432418
## 2       nox 0.1809812
## 1      crim 0.1490955
## 3     black 0.1094326
```

b) The model using lstat (percentage of lower status of the population) provides the best fit. It has the highest Adjusted $R^2$ value of approximately 0.5432, meaning it explains about 54.3% of the variance in median home values.

c) The estimated slope coefficients ($\hat{\beta}_1$) for each model represent the marginal effect of that predictor on the median home value ($medv$):

crim$(\hat{\beta}_1 = -0.415)$: Households require lower prices to live in areas with higher crime rates (a "negative amenity").

NOx$(\hat{\beta}_1 = -33.916)$: The high magnitude reflects the sensitivity of housing markets to environmental quality. Higher nitrogen oxide levels represent industrialization or traffic congestion, which negatively impacts residential utility.

black$(\hat{\beta}_1 = 0.034)$: The positive coefficient suggests a slight premium associated with the proportion of the Black population in this specific 1970s dataset, though it is the weakest predictor in terms of explanatory power.

lstat$(\hat{\beta}_1 = -0.950)$: This reflects the strong inverse relationship between neighborhood poverty levels and property values.

`lstat` is the most useful individual predictor. With an $R^2 = 0.54$, it explains over 54% of the variation in `medv`. This suggests that socioeconomic status is a primary determinant of housing market equilibrium.

`black` ($R^2 = 0.10$) and `crim` ($R^2 = 0.15$) have low explanatory power. While statistically significant ($p < 0.05$), they leave the vast majority of price variation unexplained.