Predictive Analytics Problem Set 2

746_Subhodeep Bhattacharjee

2026-02-04

Linear Regression

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

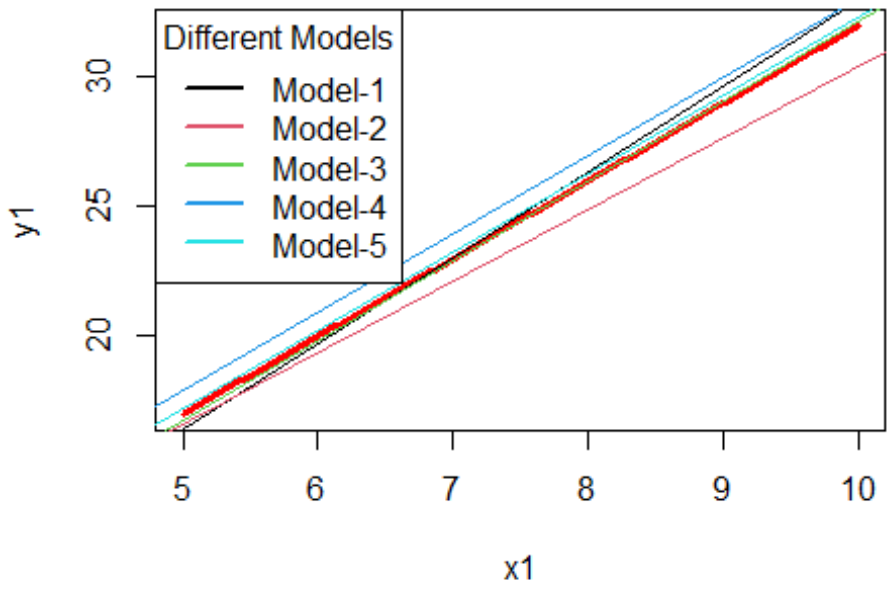
Suppose the population regression line is given by Y = 2 + 3x, while the data comes from the model y = 2 + 3x + ϵ.
- Step 1: For x in the range [5,10] graph the population regression line.
- Step 2: Generate x_i(i = 1, 2, .., n) from Uniform(5, 10) and ϵ_i(i = 1, 2, ..,n) from N(0,16). Hence, compute $y_1, y_2, .., y_n$.
- 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.

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

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

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 = i)
  sim_results$intercept[i] =coef(mod)[1]
  sim_results$slope[i]= coef(mod)[2]
  print(summary(mod)$coefficients) # detailed output
}
```

```
##                Estimate Std. Error     t value     Pr(>|t|)
## (Intercept) -0.09638929  2.8260956 -0.03410688 9.729334e-01
## x            3.30539569  0.3651861  9.05126412 5.961769e-12
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 2.792188  3.7483426 0.7449128 4.599563e-01
```

```
## x           2.761042  0.4898233 5.6368122 8.950156e-07
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 1.392997  2.7885513 0.4995416 6.196801e-01
## x           3.073267  0.3670719 8.3723847 6.062269e-11
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 2.823089  2.7354376 1.032043 3.072243e-01
## x           3.023608  0.3623615 8.344175 6.681863e-11
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 2.032506  3.1772492 0.6397063 5.254052e-01
## x           3.028097  0.4091446 7.4010440 1.789935e-09

legend("topleft",
       legend = paste(c("Model-1","Model-2","Model-3","Model-4","Model-5")),
       col = 1:5,lty = 1, lwd = 2,title = "Different Models")
```




```
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

- Red Line = Population Regression Line i.e Y = 2+ 3x
- All other lines are Least Square Regression Lines from 5 samples.

- We infer that all other lines are close to the population line but are not identical. They also differ slightly in slope and intercept.
- The population regression line is fixed because it represents a true relationship while Least Square Regression line varies because each sample is different with different random errors.

---

2. Problem to demonstrate that β$ˆ$0 and βˆ minimises RSS

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 + ϵ_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 in step 1. Assuming a linear regression of the type y_i = β_0+ βx_i + ϵ_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 − β_0 − βx_1)+ (y_2 − β_0 − βx_2) + ...(y_n − β_0 − βx_n). Find out for which combination of (β_0, β), RSS is minimum.

```
rm(list=ls())
set.seed(123)
n=50
xi = runif(n, 5, 10)
xi_std = xi - mean(xi)
eps = rnorm(n, 0, 1)
y =2 + 3 * xi_std + eps

fit = lm(y ~ xi_std)
beta0_hat = coef(fit)[1]
beta_hat = coef(fit)[2]

beta0_grid = seq(beta0_hat - 2, beta0_hat + 2, length = 51)
beta_grid = seq(beta_hat - 2, beta_hat + 2, length = 51)

RSS = matrix(, 50, 50)
for (i in 1:50) {
  for (j in 1:50) {
    RSS[i, j] = sum((y - beta0_grid[i] - beta_grid[j] * xi_std)^2)
  }
}
```

```r
which_min = which(RSS == min(RSS), arr.ind = TRUE)
which_min
```

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

```r
beta0_min = beta0_grid[which_min[1]]
beta_min = beta_grid[which_min[2]]

paste("Beta_0 min = ",beta0_min)
```

```
## [1] "Beta_0 min =  2.05618921748096"
```

```r
paste("Beta min = ",beta_min)
```

```
## [1] "Beta min =  3.07634892259978"
```

```r
print("Comparing")
```

```
## [1] "Comparing"
```

```r
paste("Beta_0 hat = ", beta0_hat)
```

```
## [1] "Beta_0 hat =  2.05618921748096"
```

```r
paste("Beta hat = ", beta_hat)
```

```
## [1] "Beta hat =  3.07634892259978"
```

Interpretation

- Mean Centering is done because it simplifies interpretation and ensures that the intercept estimate is stable and uncorrelated with the slope.
- By evaluating the residual sum of squares over a grid of parameter values, it is observed that RSS achieves its minimum at the least squares estimates $\hat{\beta}_0$ and $\hat{\beta}$. This confirms that the ordinary least squares estimators minimize the RSS, validating the fundamental principle of linear regression estimation.

---

3. Problem to demonstrate that least square estimators are unbiased

Step 1: Generate xi(i = 1, 2, .., n) from Uniform(0, 1), ϵ_i(i = 1, 2, .., n) from N(0, 1) and hence generate y using $y_i = \beta_0 + \beta x_i + \epsilon_i$. (Take β_0= 2, β = 3).
Step 2: On the basis of the data (x_i, y_i)(i = 1, 2, .., 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.

```r
rm(list=ls())
set.seed(123)
n=50
R=1000

beta0 = 2
beta1 = 3

beta0_hat = numeric(R)
beta1_hat = numeric(R)

for (r in 1:R) {
  x = runif(n, 0, 1)
  eps = rnorm(n, 0, 1)
  y = beta0 + beta1 * x + eps
  fit = lm(y ~ x)
  beta0_hat[r] = coef(fit)[1]
  beta1_hat[r] = coef(fit)[2]
}

mean(beta0_hat)
```

```
## [1] 2.013053
```

```r
mean(beta1_hat)
```

```
## [1] 2.982112
```

```r
c(True_beta0 = beta0,
  Estimated_beta0 = mean(beta0_hat),
  True_beta = beta1,
  Estimated_beta = mean(beta1_hat))
```

```
##      True_beta0 Estimated_beta0       True_beta  Estimated_beta
##        2.000000        2.013053        3.000000        2.982112
```

Interpretation

Using repeated sampling, the least squares estimates of the intercept and slope were obtained 1000 times. The average of the estimated intercepts is approximately equal to the true intercept value of 2, and the average of the estimated slopes is approximately equal to the true slope value of 3. Hence, the least squares estimators $\hat{\beta}_0$ and $\hat{\beta}$ are unbiased estimators of β_0 and β, respectively.

---

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.

```
library(MASS)
data(Boston)

str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524
0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Ans a)

```
m1 = lm(medv ~ crim, data = Boston)
m2 = lm(medv ~ nox, data = Boston)
m3 = lm(medv ~ black, data = Boston)
m4 = lm(medv ~ lstat, data = Boston)

results = data.frame(
  Model = c("medv ~ crim", "medv ~ nox", "medv ~ black", "medv ~ lstat"),
```

```
  Intercept = c(coef(m1)[1], coef(m2)[1], coef(m3)[1], coef(m4)[1]),

  Slope = c(coef(m1)[2], coef(m2)[2], coef(m3)[2], coef(m4)[2]),

  R_squared = c(summary(m1)$r.squared,
                summary(m2)$r.squared,
                summary(m3)$r.squared,
                summary(m4)$r.squared),

  Adj_R_squared = c(summary(m1)$adj.r.squared,
                    summary(m2)$adj.r.squared,
                    summary(m3)$adj.r.squared,
                    summary(m4)$adj.r.squared),

  p_value = c(summary(m1)$coefficients[2,4],
              summary(m2)$coefficients[2,4],
              summary(m3)$coefficients[2,4],
              summary(m4)$coefficients[2,4])
)

results
```

```
##              Model Intercept        Slope R_squared Adj_R_squared
p_value
## crim   medv ~ crim  24.03311  -0.41519028 0.1507805     0.1490955
1.173987e-19
## nox     medv ~ nox  41.34587 -33.91605501 0.1826030     0.1809812
7.065042e-24
## black medv ~ black  10.55103   0.03359306 0.1111961     0.1094326
1.318113e-14
## lstat medv ~ lstat  34.55384  -0.95004935 0.5441463     0.5432418
5.081103e-88
```

Ans b)
The best fit model is Model 4 which is medv ~ lstat. It has a very high R^2 / adjusted R^2 value and very small p-value holding a very strong statistical explanatory power.

Ans c)

1. medv ~ crim

- Slope < 0
- Higher crime rate → lower house prices

- Weak explanatory power (low R^2)
- Useful, but not strong alone

2. medv ~ nox

- Slope < 0
- Higher pollution → lower house prices
- Moderate R^2
- Important environmental factor

3. medv ~ black

- Slope > 0
- Higher proportion associated with higher median values
- Low to moderate R^2
- Weak predictor when used alone

4. medv ~ lstat

- Slope < 0
- Strong negative relationship
- Explains over 50% of variability
- Most powerful single predictor

Among the four simple linear regression models, the model with percentage of lower status population (lstat) as the predictor provides the best fit, as it has the highest R-squared value. This indicates that lstat alone explains a large proportion of the variability in median house values. Crime rate and nitrogen oxides concentration also show negative relationships with housing prices but explain comparatively less variation. The proportion of blacks is the weakest predictor when considered individually. Hence, lstat is the most useful predictor among the four in explaining median house values.