Assignment_2

PROTYAY CHATTERJEE

2026-02-04

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 xi(i = 1, 2, .., n) from Uniform(5, 10) and εi(i = 1, 2, .., n) from $N(0, 4^2)$. Hence, compute y1, y2, .., yn. Step 3: On the basis of the data (xi, yi)(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)
x=seq(5,10,0.025)
y=2+3*x
y
```

```
##   [1] 17.000 17.075 17.150 17.225 17.300 17.375 17.450 17.525 17.600
17.675
##  [11] 17.750 17.825 17.900 17.975 18.050 18.125 18.200 18.275 18.350
18.425
##  [21] 18.500 18.575 18.650 18.725 18.800 18.875 18.950 19.025 19.100
19.175
##  [31] 19.250 19.325 19.400 19.475 19.550 19.625 19.700 19.775 19.850
19.925
##  [41] 20.000 20.075 20.150 20.225 20.300 20.375 20.450 20.525 20.600
20.675
##  [51] 20.750 20.825 20.900 20.975 21.050 21.125 21.200 21.275 21.350
21.425
##  [61] 21.500 21.575 21.650 21.725 21.800 21.875 21.950 22.025 22.100
22.175
##  [71] 22.250 22.325 22.400 22.475 22.550 22.625 22.700 22.775 22.850
22.925
##  [81] 23.000 23.075 23.150 23.225 23.300 23.375 23.450 23.525 23.600
23.675
##  [91] 23.750 23.825 23.900 23.975 24.050 24.125 24.200 24.275 24.350
24.425
## [101] 24.500 24.575 24.650 24.725 24.800 24.875 24.950 25.025 25.100
25.175
## [111] 25.250 25.325 25.400 25.475 25.550 25.625 25.700 25.775 25.850
25.925
## [121] 26.000 26.075 26.150 26.225 26.300 26.375 26.450 26.525 26.600
26.675
```

```
## [131] 26.750 26.825 26.900 26.975 27.050 27.125 27.200 27.275 27.350
27.425
## [141] 27.500 27.575 27.650 27.725 27.800 27.875 27.950 28.025 28.100
28.175
## [151] 28.250 28.325 28.400 28.475 28.550 28.625 28.700 28.775 28.850
28.925
## [161] 29.000 29.075 29.150 29.225 29.300 29.375 29.450 29.525 29.600
29.675
## [171] 29.750 29.825 29.900 29.975 30.050 30.125 30.200 30.275 30.350
30.425
## [181] 30.500 30.575 30.650 30.725 30.800 30.875 30.950 31.025 31.100
31.175
## [191] 31.250 31.325 31.400 31.475 31.550 31.625 31.700 31.775 31.850
31.925
## [201] 32.000

#1
plot(x,y,type='l')
#2
x_i=runif(50,5,10)
x_i

##  [1] 6.437888 8.941526 7.044885 9.415087 9.702336 5.227782 7.640527
9.462095
##  [9] 7.757175 7.283074 9.784167 7.266671 8.387853 7.863167 5.514623
9.499125
## [17] 6.230439 5.210298 6.639604 9.772518 9.447697 8.464017 8.202534
9.971349
## [25] 8.278529 8.542652 7.720330 7.970710 6.445799 5.735568 9.815121
9.511495
## [33] 8.453526 8.977337 5.123068 7.388980 8.792298 6.082040 6.590905
6.158129
## [41] 5.714000 7.072732 7.068622 6.844227 5.762224 5.694030 6.165170
7.329812
## [49] 6.329863 9.289139

ei=rnorm(50,0,4)
ei

##  [1] -6.7467732  3.3511482  0.6134925 -4.5525477  5.0152597  1.7058569
##  [7] -1.1802859  3.5805026  3.5125340  3.2863243  2.7545610  2.2156706
## [13] -0.2476468 -1.2238507 -1.5218840 -2.7788279 -0.8316691 -5.0615854
## [19]  8.6758239  4.8318480 -4.4924343 -1.6115393 -1.8666214  3.1198605
## [25] -0.3334763  1.0132741 -0.1141870 -0.1714818  5.4744091 -0.9030839
## [31]  6.0658824 -6.1950112  2.3384550  0.4954170  0.8637663  1.5185579
## [37] -2.0092938 -1.3328295 -4.0743015 -4.2871649  1.2141146  1.7928391
## [43]  0.2120169  3.6890699  8.2003387 -1.9641247 -9.2366755  4.0229541
## [49] -2.8368031 -2.7520345

y_i=2+3*x_i+ei
y_i
```

```
##  [1] 14.56689 32.17573 23.74815 25.69271 36.12227 19.38920 23.74130
33.96679
##  [9] 28.78406 27.13555 34.10706 26.01568 26.91591 24.36565 17.02199
27.71855
## [17] 19.85965 12.56931 30.59463 36.14940 25.85066 25.78051 24.74098
35.03391
## [25] 26.50211 28.64123 25.04680 25.74065 26.81181 18.30362 37.51125
24.33947
## [33] 29.69903 29.42743 18.23297 25.68550 26.36760 18.91329 17.69841
16.18722
## [41] 20.35611 25.01103 23.41788 26.22175 27.48701 17.11797 11.25884
28.01239
## [49] 18.15279 27.11538
```

```r
points(x_i,y_i,col='red')
#3

linreg=lm(y_i~x_i)
abline(linreg,col='blue')
coef(linreg)
```

```
## (Intercept)         x_i
## -0.09638929  3.30539569
```

```r
summary(linreg)
```

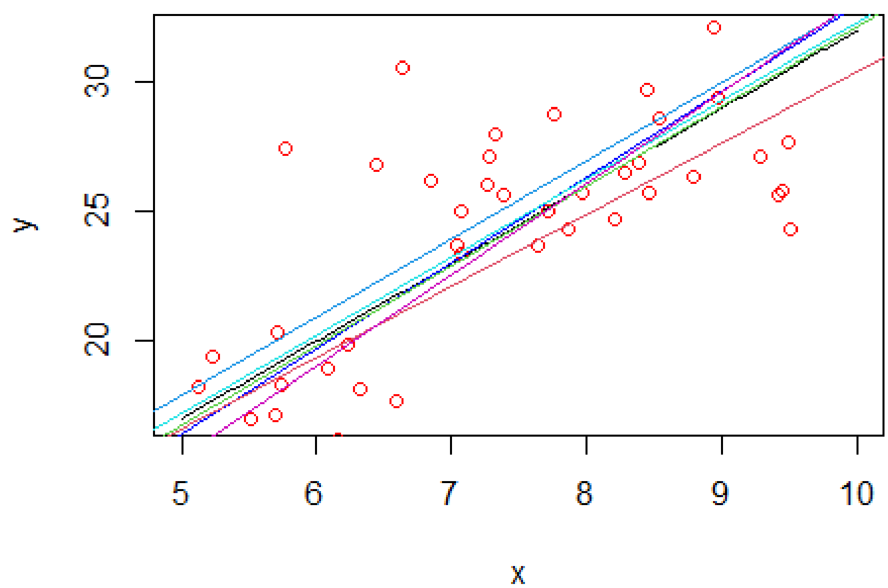
```
##
## Call:
## lm(formula = y_i ~ x_i)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0231 -2.2314 -0.2627  2.1970  8.7445
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.09639    2.82610  -0.034    0.973
## x_i          3.30540    0.36519   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
```

```r
#4

## beta not is 2 however the intrcept is -0.0963... which are not similar


beta_hat=c()
```

```
for(i in 1:5){
  x_i=runif(50,5,10)
  ei=rnorm(50,0,4)
  y_i=2+3*x_i+ei
  linreg=lm(y_i~x_i)
  beta_hat=cbind(beta_hat,coef(linreg))
  abline(linreg,col=i+1)
  coef(linreg)
}
```




```
beta.hat=as.data.frame(beta_hat)
beta.hat

##                    V1       V2       V3       V4        V5
## (Intercept) 2.792188 1.392997 2.823089 2.032506 -2.107763
## x_i         2.761042 3.073267 3.023608 3.028097  3.530691
```

2. Problem to demonstrate that β̂0 and β̂ minimises RSS

Step 1: Generate xi from Uniform(5, 10) and mean centre the values. Generate εi from N(0, 1). Calculate yi = 2+3xi+εi, i = 1,2,.., n. Take n=50 and seed=123. Step 2: Now imagine that you only have the data on (xi, yi), 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 yi = β0 + βxi + εi, and based on these data (xi, yi), i = 1, 2, .., n, obtain the least squares estimates of β0 and β. Step 3: Take a large number of grid values of (β0, β) that also include the least squares estimates obtained from step 2. Compute the RSS for each parametric choice of (β0, β), where RSS = $(y1 - \beta 0 - \beta x1)^2 + (y2 - \beta 0 -$

$\beta x2)^2$ + ….$(yn - \beta 0 - \beta xn)^2$. Find out for which combination of (β0, β), RSS is minimum.

```
set.seed(123)
x_ii=runif(50,5,10)
x_ii

##  [1] 6.437888 8.941526 7.044885 9.415087 9.702336 5.227782 7.640527
9.462095
##  [9] 7.757175 7.283074 9.784167 7.266671 8.387853 7.863167 5.514623
9.499125
## [17] 6.230439 5.210298 6.639604 9.772518 9.447697 8.464017 8.202534
9.971349
## [25] 8.278529 8.542652 7.720330 7.970710 6.445799 5.735568 9.815121
9.511495
## [33] 8.453526 8.977337 5.123068 7.388980 8.792298 6.082040 6.590905
6.158129
## [41] 5.714000 7.072732 7.068622 6.844227 5.762224 5.694030 6.165170
7.329812
## [49] 6.329863 9.289139

eii=rnorm(50,0,1)
eii

##  [1] -1.68669331  0.83778704  0.15337312 -1.13813694  1.25381492
0.42646422
##  [7] -0.29507148  0.89512566  0.87813349  0.82158108  0.68864025
0.55391765
## [13] -0.06191171 -0.30596266 -0.38047100 -0.69470698 -0.20791728 -
1.26539635
## [19]  2.16895597  1.20796200 -1.12310858 -0.40288484 -0.46665535
0.77996512
## [25] -0.08336907  0.25331851 -0.02854676 -0.04287046  1.36860228 -
0.22577099
## [31]  1.51647060 -1.54875280  0.58461375  0.12385424  0.21594157
0.37963948
## [37] -0.50232345 -0.33320738 -1.01857538 -1.07179123  0.30352864
0.44820978
## [43]  0.05300423  0.92226747  2.05008469 -0.49103117 -2.30916888
1.00573852
## [49] -0.70920076 -0.68800862

y_ii=2+3*x_ii+eii
y_ii

##  [1] 19.62697 29.66236 23.28803 29.10712 32.36082 18.10981 24.62651
31.28141
##  [9] 26.14966 24.67080 32.04114 24.35393 27.10165 25.28354 18.16340
29.80267
## [17] 20.48340 16.36550 24.08777 32.52552 29.21998 26.98917 26.14095
32.69401
```

```
## [25] 26.75222 27.88128 25.13244 25.86926 22.70600 18.98093 32.96183
28.98573
## [33] 27.94519 29.05587 17.58515 24.54658 27.87457 19.91291 20.75414
19.40260
## [41] 19.44553 23.66640 23.25887 23.45495 21.33676 18.59106 18.18634
24.99518
## [49] 20.28039 29.17941

model= lm(y_ii ~ x_ii)

# Least squares estimates
beta0_hat=coef(model) [1]
beta_hat=coef(model)[2]

beta0_hat

## (Intercept)
##    1.475903

beta_hat

##     x_ii
## 3.076349

beta0_grid= seq(beta0_hat-2, beta0_hat + 2, length.out = 100)
beta_grid = seq(beta_hat-2, beta_hat + 2, length.out = 100)

RSS <- matrix(NA, nrow = length(beta0_grid), ncol = length(beta_grid))

# Compute RSS for each (beta0, beta)
for (i in 1:length(beta0_grid)) {
  for (j in 1:length(beta_grid)) {
    RSS[i, j] <- sum((y_ii - beta0_grid[i] - beta_grid[j] * x_ii)^2)
  }
}
which(RSS == min(RSS), arr.ind=TRUE)

##      row col
## [1,]  47  51

# Find minimum RSS
min_RSS <- min(RSS)
index <- which(RSS == min_RSS, arr.ind = TRUE)

beta0_min <- beta0_grid[index[1]]
beta_min  <- beta_grid[index[2]]

beta0_min

## [1] 1.334489
```

```
beta_min
```

```
## [1] 3.096551
```

3. Problem to demonstrate that least square estimators are unbiased

Step 1: Generate xi(i = 1, 2, .., n) from Uniform(0, 1), εi(i = 1, 2, .., n) from N(0, 1) and hence generate y using yi = β0 + βxi + εi. (Take β0 = 2, β = 3).

Step 2: On the basis of the data (xi, yi)(i = 1, 2, .., n) generated in Step 1, obtain the least square estimates of β0 and β. Repeat Steps 1-2, R = 1000 times. In each simulation obtain βˆ0 and βˆ. Finally,the least-square estimates will be given by the average of these estimated values. Compare these with the true β0 and β and comment. Take n = 50 and seed=123.

```
#3
set.seed(123)
x_3=runif(50,0,1)
x_3
```

```
##  [1] 0.28757752 0.78830514 0.40897692 0.88301740 0.94046728 0.04555650
##  [7] 0.52810549 0.89241904 0.55143501 0.45661474 0.95683335 0.45333416
## [13] 0.67757064 0.57263340 0.10292468 0.89982497 0.24608773 0.04205953
## [19] 0.32792072 0.95450365 0.88953932 0.69280341 0.64050681 0.99426978
## [25] 0.65570580 0.70853047 0.54406602 0.59414202 0.28915974 0.14711365
## [31] 0.96302423 0.90229905 0.69070528 0.79546742 0.02461368 0.47779597
## [37] 0.75845954 0.21640794 0.31818101 0.23162579 0.14280002 0.41454634
## [43] 0.41372433 0.36884545 0.15244475 0.13880606 0.23303410 0.46596245
## [49] 0.26597264 0.85782772
```

```
e_3=rnorm(50,0,1)
e_3
```

```
##  [1] -1.68669331  0.83778704  0.15337312 -1.13813694  1.25381492
0.42646422
##  [7] -0.29507148  0.89512566  0.87813349  0.82158108  0.68864025
0.55391765
## [13] -0.06191171 -0.30596266 -0.38047100 -0.69470698 -0.20791728 -
1.26539635
## [19]  2.16895597  1.20796200 -1.12310858 -0.40288484 -0.46665535
0.77996512
## [25] -0.08336907  0.25331851 -0.02854676 -0.04287046  1.36860228 -
0.22577099
## [31]  1.51647060 -1.54875280  0.58461375  0.12385424  0.21594157
0.37963948
## [37] -0.50232345 -0.33320738 -1.01857538 -1.07179123  0.30352864
0.44820978
## [43]  0.05300423  0.92226747  2.05008469 -0.49103117 -2.30916888
1.00573852
## [49] -0.70920076 -0.68800862
```

```r
y_3=2+3*x_3+e_3
model_3= lm(y_3 ~ x_3)
model_3

##
## Call:
## lm(formula = y_3 ~ x_3)
##
## Coefficients:
## (Intercept)          x_3
##       1.858        3.382

set.seed(123)

n= 50
R= 1000

beta0_hat <- numeric(R)
beta_hat  <- numeric(R)

for (r in 1:R) {

  x_3 <- runif(n, 0, 1)
  e_3 <- rnorm(n, 0, 1)     # using our error term
  y_3 <- 2 + 3 * x_3 + e_3

  model_3 <- lm(y_3 ~ x_3)

  beta0_hat[r] <- coef(model_3)[1]
  beta_hat[r]  <- coef(model_3)[2]
}
mean(beta0_hat)

## [1] 2.013053

mean(beta_hat)

## [1] 2.982112

var(beta0_hat)

## [1] 0.07969639

var(beta_hat)

## [1] 0.2360544
```

This demonstrate that the Least Squares Estimators are unbiased.The average of the 1,000 estimated coefficients ($\hat{\beta}_0$ and $\hat{\beta}_1$) is extremely close to the true parameter values ($\beta_0 = 2$ and $\beta_1 = 3$). The small differences observed in the table are due to sampling variability and would approach zero as the number of simulations (R) increases towards infinity.

Question 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 popu- lation 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)
x1=Boston
y=Boston$medv#response
x=data.frame(x1$crim,x1$nox,x1$black,x1$lstat)#predictor
y

##   [1] 24.0 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 15.0 18.9 21.7
20.4 18.2
##  [16] 19.9 23.1 17.5 20.2 18.2 13.6 19.6 15.2 14.5 15.6 13.9 16.6 14.8
18.4 21.0
##  [31] 12.7 14.5 13.2 13.1 13.5 18.9 20.0 21.0 24.7 30.8 34.9 26.6 25.3
24.7 21.2
##  [46] 19.3 20.0 16.6 14.4 19.4 19.7 20.5 25.0 23.4 18.9 35.4 24.7 31.6
23.3 19.6
##  [61] 18.7 16.0 22.2 25.0 33.0 23.5 19.4 22.0 17.4 20.9 24.2 21.7 22.8
23.4 24.1
##  [76] 21.4 20.0 20.8 21.2 20.3 28.0 23.9 24.8 22.9 23.9 26.6 22.5 22.2
23.6 28.7
##  [91] 22.6 22.0 22.9 25.0 20.6 28.4 21.4 38.7 43.8 33.2 27.5 26.5 18.6
19.3 20.1
## [106] 19.5 19.5 20.4 19.8 19.4 21.7 22.8 18.8 18.7 18.5 18.3 21.2 19.2
20.4 19.3
## [121] 22.0 20.3 20.5 17.3 18.8 21.4 15.7 16.2 18.0 14.3 19.2 19.6 23.0
18.4 15.6
## [136] 18.1 17.4 17.1 13.3 17.8 14.0 14.4 13.4 15.6 11.8 13.8 15.6 14.6
17.8 15.4
## [151] 21.5 19.6 15.3 19.4 17.0 15.6 13.1 41.3 24.3 23.3 27.0 50.0 50.0
50.0 22.7
## [166] 25.0 50.0 23.8 23.8 22.3 17.4 19.1 23.1 23.6 22.6 29.4 23.2 24.6
29.9 37.2
## [181] 39.8 36.2 37.9 32.5 26.4 29.6 50.0 32.0 29.8 34.9 37.0 30.5 36.4
31.1 29.1
## [196] 50.0 33.3 30.3 34.6 34.9 32.9 24.1 42.3 48.5 50.0 22.6 24.4 22.5
24.4 20.0
```

```
## [211] 21.7 19.3 22.4 28.1 23.7 25.0 23.3 28.7 21.5 23.0 26.7 21.7 27.5
30.1 44.8
## [226] 50.0 37.6 31.6 46.7 31.5 24.3 31.7 41.7 48.3 29.0 24.0 25.1 31.5
23.7 23.3
## [241] 22.0 20.1 22.2 23.7 17.6 18.5 24.3 20.5 24.5 26.2 24.4 24.8 29.6
42.8 21.9
## [256] 20.9 44.0 50.0 36.0 30.1 33.8 43.1 48.8 31.0 36.5 22.8 30.7 50.0
43.5 20.7
## [271] 21.1 25.2 24.4 35.2 32.4 32.0 33.2 33.1 29.1 35.1 45.4 35.4 46.0
50.0 32.2
## [286] 22.0 20.1 23.2 22.3 24.8 28.5 37.3 27.9 23.9 21.7 28.6 27.1 20.3
22.5 29.0
## [301] 24.8 22.0 26.4 33.1 36.1 28.4 33.4 28.2 22.8 20.3 16.1 22.1 19.4
21.6 23.8
## [316] 16.2 17.8 19.8 23.1 21.0 23.8 23.1 20.4 18.5 25.0 24.6 23.0 22.2
19.3 22.6
## [331] 19.8 17.1 19.4 22.2 20.7 21.1 19.5 18.5 20.6 19.0 18.7 32.7 16.5
23.9 31.2
## [346] 17.5 17.2 23.1 24.5 26.6 22.9 24.1 18.6 30.1 18.2 20.6 17.8 21.7
22.7 22.6
## [361] 25.0 19.9 20.8 16.8 21.9 27.5 21.9 23.1 50.0 50.0 50.0 50.0 50.0
13.8 13.8
## [376] 15.0 13.9 13.3 13.1 10.2 10.4 10.9 11.3 12.3  8.8  7.2 10.5  7.4
10.2 11.5
## [391] 15.1 23.2  9.7 13.8 12.7 13.1 12.5  8.5  5.0  6.3  5.6  7.2 12.1
8.3  8.5
## [406]  5.0 11.9 27.9 17.2 27.5 15.0 17.2 17.9 16.3  7.0  7.2  7.5 10.4
8.8  8.4
## [421] 16.7 14.2 20.8 13.4 11.7  8.3 10.2 10.9 11.0  9.5 14.5 14.1 16.1
14.3 11.7
## [436] 13.4  9.6  8.7  8.4 12.8 10.5 17.1 18.4 15.4 10.8 11.8 14.9 12.6
14.1 13.0
## [451] 13.4 15.2 16.1 17.8 14.9 14.1 12.7 13.5 14.9 20.0 16.4 17.7 19.5
20.2 21.4
## [466] 19.9 19.0 19.1 19.1 20.1 19.9 19.6 23.2 29.8 13.8 13.3 16.7 12.0
14.6 21.4
## [481] 23.0 23.7 25.0 21.8 20.6 21.2 19.1 20.6 15.2  7.0  8.1 13.6 20.1
21.8 24.5
## [496] 23.1 19.7 18.3 21.2 17.5 16.8 22.4 20.6 23.9 22.0 11.9
```

```r
model1=lm(y~x1$crim)
model1
```

```
##
## Call:
## lm(formula = y ~ x1$crim)
##
## Coefficients:
## (Intercept)      x1$crim
##     24.0331      -0.4152
```

```
summary(model1)

##
## Call:
## lm(formula = y ~ x1$crim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.957  -5.449  -2.007   2.512  29.800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## x1$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

model2=lm(y~x1$nox)
model2

##
## Call:
## lm(formula = y ~ x1$nox)
##
## Coefficients:
## (Intercept)       x1$nox
##       41.35       -33.92

model3=lm(y~x1$black)
model3

##
## Call:
## lm(formula = y ~ x1$black)
##
## Coefficients:
## (Intercept)     x1$black
##    10.55103      0.03359

model4=lm(y~x1$lstat)
model4

##
## Call:
## lm(formula = y ~ x1$lstat)
##
## Coefficients:
```

```
## (Intercept)     x1$lstat
##       34.55        -0.95
```

```r
models=c("model of y vs crim","model of y vs nox","model of y vs
black","model of y vs lstat")
beta_hat_0=c(coef(model1)[1],coef(model2)[1],coef(model3)[1],coef(model4)[1])
beta_hat=c(coef(model1)[2],coef(model2)[2],coef(model3)[2],coef(model4)[2])
Data=data.frame(models,beta_hat_0,beta_hat)
Data
```

```
##                          models beta_hat_0     beta_hat
## x1$crim   model of y vs crim   24.03311  -0.41519028
## x1$nox     model of y vs nox   41.34587 -33.91605501
## x1$black model of y vs black   10.55103   0.03359306
## x1$lstat model of y vs lstat   34.55384  -0.95004935
```

```r
summary(model1)
```

```
## 
## Call:
## lm(formula = y ~ x1$crim)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -16.957  -5.449  -2.007   2.512  29.800 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 24.03311    0.40914   58.74   <2e-16 ***
## x1$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
```

```r
summary(model2)
```

```
## 
## Call:
## lm(formula = y ~ x1$nox)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -13.691  -5.121  -2.161   2.959  31.310 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   41.346      1.811   22.83   <2e-16 ***
## x1$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
```

```r
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x1$black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.884  -4.862  -1.684   2.932  27.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.551034   1.557463   6.775 3.49e-11 ***
## x1$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
```

```r
summary(model4)
```

```
##
## Call:
## lm(formula = y ~ x1$lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## x1$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
```

```r
extract <- function(m){
  c(
```

```r
    Intercept = coef(m)[1],
    Slope = coef(m)[2],
    R2 = summary(m)$r.squared,
    p_value = summary(m)$coefficients[2,4]
  )
}

results <- rbind(
  crim = extract(model1),
  nox = extract(model2),
  black = extract(model3),
  lstat = extract(model4)
)

results

##       Intercept.(Intercept) Slope.x1$crim        R2      p_value
## crim               24.03311   -0.41519028 0.1507805 1.173987e-19
## nox                41.34587  -33.91605501 0.1826030 7.065042e-24
## black              10.55103    0.03359306 0.1111961 1.318113e-14
## lstat              34.55384   -0.95004935 0.5441463 5.081103e-88
```

The model with lstat as predictor gives the best fit.

Highest R-squared (around 0.54, much larger than others), ,Very small p-value (≈ 0),Strong linear relationship with medv,Lowest residual standard error

Comparing coefficients and usefulness of predictors:

Crime rate (crim):

Negative coefficient → higher crime lowers home value

Statistically significant

Explains only a small portion of variability

Useful, but not a strong standalone predictor

Nitrogen oxides (nox)

Negative coefficient → more pollution lowers prices

Stronger effect than crime

Still limited explanatory power alone

Proportion of blacks (black)

Positive coefficient

Statistically significant, but low R^2

Weak predictor by itself

Lower status population (lstat)

Strong negative coefficient

Extremely significant

Largest magnitude effect

Best single predictor among the four