LAB - 3 MLR: Part 2

PrabhTalwar

2022-10-16

MLR using simulated data

```
set.seed(11235)
n = 100 # sample size
p = 4
beta_0 = 4
beta_1 = -1
beta_2 = 5
beta_3 = 2
sigma = 4
```

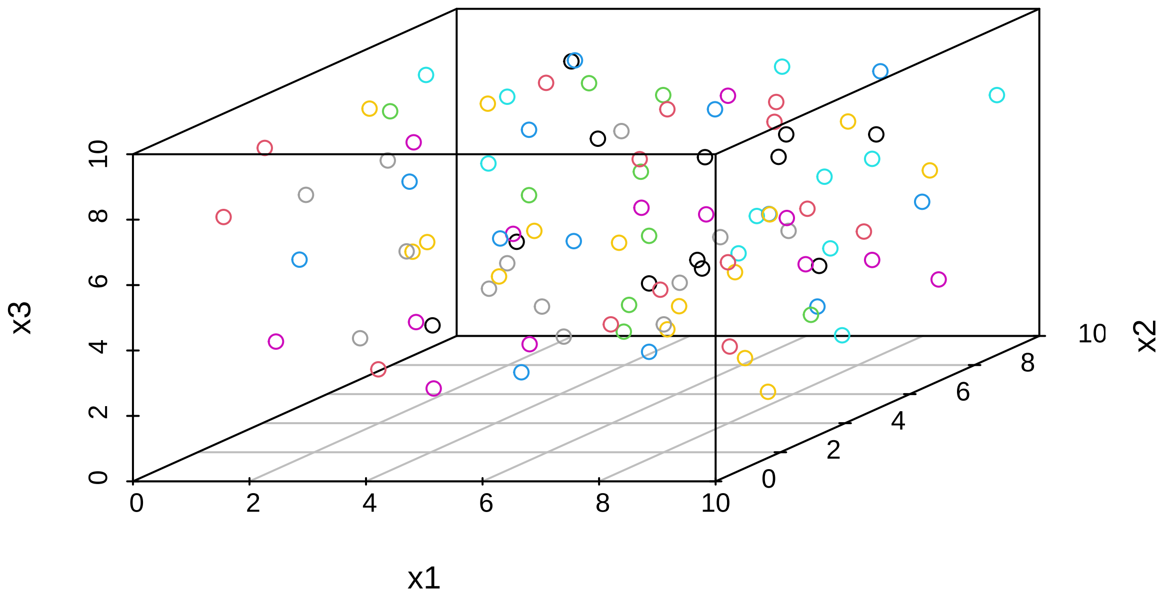
```
# Creating X Matrix and C Matrix
x0 = rep(1, n)
x1 = sample(seq(1, 10, length = n))
x2 = sample(seq(1, 10, length = n))
x3 = sample(seq(1, 10, length = n))

X = cbind(x0, x1, x2, x3)
C = solve(t(X) %*% X)
```

```
# Simulating the response according the model
eps = rnorm(n, mean = 0, sd = sigma)
y = beta_0 + beta_1 * x1 + beta_2 * x2 + beta_3 * x3 + eps
sim_data = data.frame(x1, x2, x3, y)
```

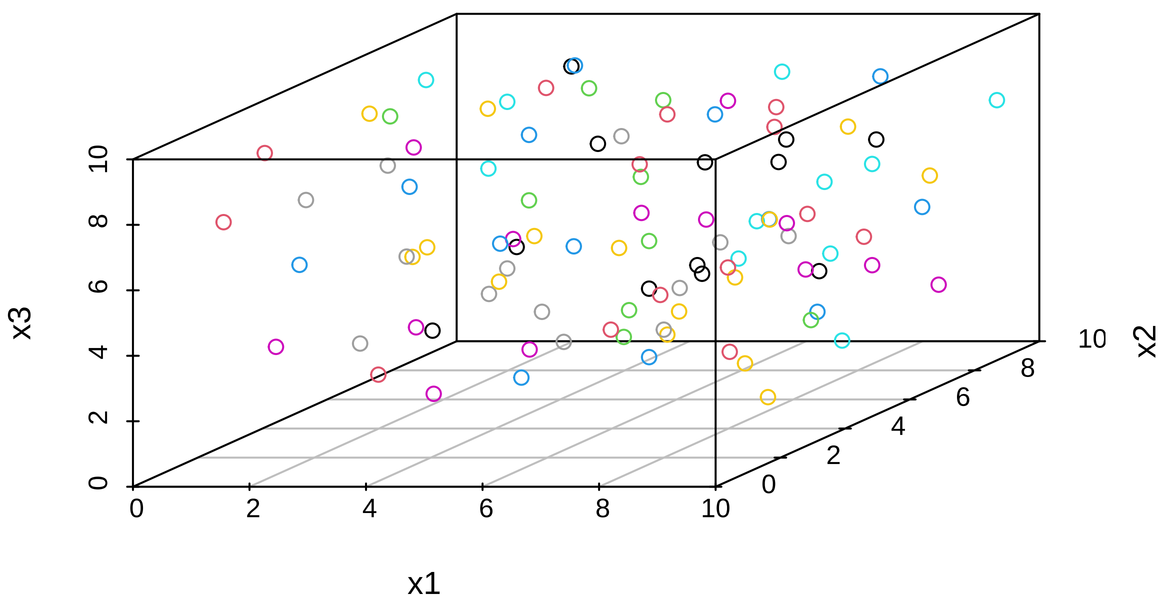
```
# Plotting the data
library(scatterplot3d)
```

```
## Warning: package 'scatterplot3d' was built under R version 4.1.3
```

```
scatterplot3d(sim_data)
```




```
sim.lm <- lm(y ~ x1 + x2 + x3, data = sim_data)
sim3d <- scatterplot3d(sim_data)
```




```
# Computing the least squares values

(beta_hat = C %*% t(X) %*% y)
```

```
##          [,1]
## x0  5.220492
## x1 -1.199235
## x2  4.847678
## x3  2.100326
```

```
c(beta_0, beta_1, beta_2, beta_3)
```

```
## [1]  4 -1  5  2
```

```
# Calculating the fitted values in order to calculate se
y_hat = X %*% beta_hat
(s_e = sqrt(sum((y - y_hat) ^ 2) / (n - p)))
```

```
## [1] 3.794345
```

```
summary(lm(y ~ x1 + x2 + x3, data = sim_data))$sigma
```

```
## [1] 3.794345
```

```r
# Simulating in order to obtain an empirical distribution

C = solve(t(X) %*% X)

C[4, 4]
```

```
## [1] 0.001495816
```

```r
C[3 + 1, 3 + 1]
```

```
## [1] 0.001495816
```

```r
sigma ^ 2 * C[3 + 1, 3 + 1]
```

```
## [1] 0.02393306
```

```r
# performing the simulation a large number of times

num_sims = 10000
beta_hat_3 = rep(0, num_sims)
for(i in 1:num_sims) {
  eps = rnorm(n, mean = 0 , sd = sigma)
  sim_data$y = beta_0 * x0 + beta_1 * x1 + beta_2 * x2 + beta_3 * x3 + eps
  fit = lm(y ~ x1 + x2 + x3, data = sim_data)
  beta_hat_3[i] = coef(fit)[4]
  }
```

```r
# mean of the simulated values

mean(beta_hat_3)
```

```
## [1] 2.000519
```

```r
beta_3
```

```
## [1] 2
```

```r
#  variance of the simulated values

var(beta_hat_3)
```

```
## [1] 0.02424589
```

```r
sigma ^ 2 * C[3 + 1, 3 + 1]
```

```
## [1] 0.02393306
```

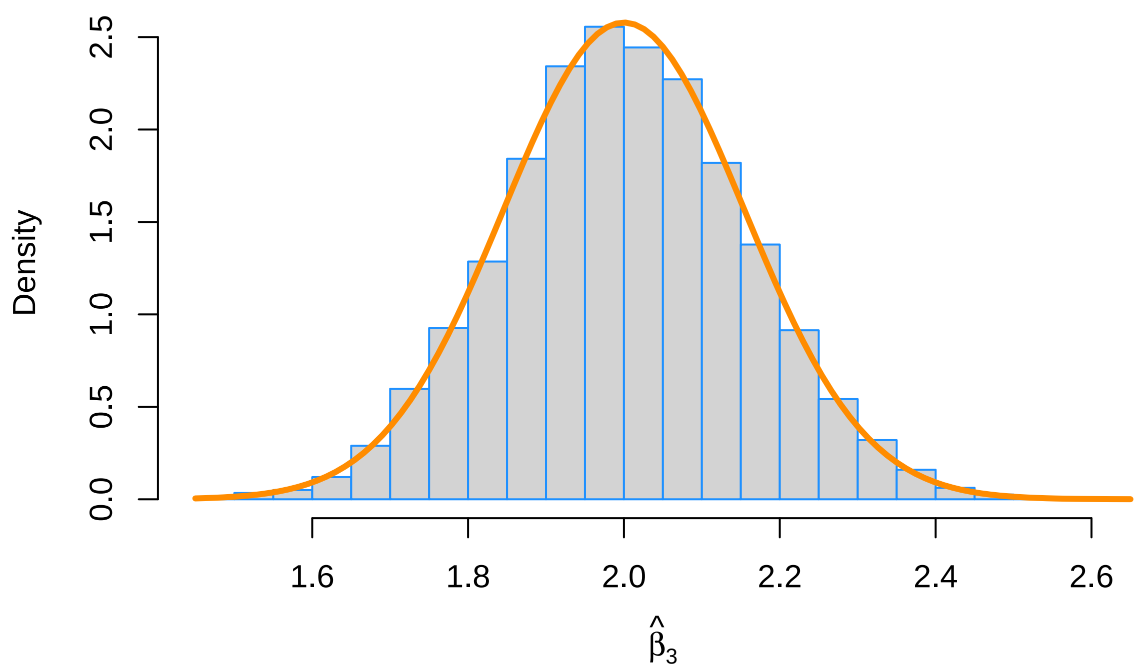
```r
#  sd of the simulated values

sd(beta_hat_3)
```

```
## [1] 0.1557109
```

```r
sqrt(sigma ^ 2 * C[3 + 1, 3 + 1])
```

```
## [1] 0.1547031
```

```r
# ploting a histogram of the simulated values

hist(beta_hat_3, prob = TRUE, breaks = 20,
     xlab = expression(hat(beta)[3]),
     main = "",
     border = "dodgerblue")

curve(dnorm(x, mean = beta_3, sd = sqrt(sigma ^ 2 * C[3 + 1, 3 + 1])),
      col = "darkorange", add = TRUE, lwd = 3)
```




```r
# verifying the 68-95-99.7 rule

sd_bh3 = sqrt(sigma ^ 2 * C[3 + 1, 3 + 1])
```

```r
# We expect these to be: 0.68, 0.95, 0.997

mean(beta_3 - 1 * sd_bh3 < beta_hat_3 & beta_hat_3 < beta_3 + 1 * sd_bh3)
```

```
## [1] 0.6778
```

```r
mean(beta_3 - 2 * sd_bh3 < beta_hat_3 & beta_hat_3 < beta_3 + 2 * sd_bh3)
```

```
## [1] 0.9525
```

```r
mean(beta_3 - 3 * sd_bh3 < beta_hat_3 & beta_hat_3 < beta_3 + 3 * sd_bh3)
```

```
## [1] 0.9972
```