Linea Regression With Matrices and LM Model in R (LaTeX also incorporated)

Joshua Mwangi Maina

Simple Linear Regression

```r
X1 <- c(30.2, 32.8, 32.9, 35.1, 42.3, 45.5, 46)
y <- c(6.8, 10.1, 14.3, 19.3, 10.2, 20, 23.7)
```

Step 1: Create a data frame

```r
data <- data.frame(y, X1)
```

Step 2: Fit the model

```r
model <- lm(y ~ X1, data = data)
summary(model)
```

```
Call:
lm(formula = y ~ X1, data = data)

Residuals:
      1       2       3       4       5       6       7 
-3.2331 -1.5967  2.5393  6.1316 -7.5754  0.1771  3.5572 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  -9.2907    11.9629  -0.777   0.4725  
X1            0.6399     0.3122   2.050   0.0957 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.042 on 5 degrees of freedom
Multiple R-squared:  0.4565,	Adjusted R-squared:  0.3479 
F-statistic:   4.2 on 1 and 5 DF,  p-value: 0.0957
```

Step 3: Predict y when x1 = 40

```r
new_data <- data.frame(X1 = 40)
predicted_y <- predict(model, newdata = new_data)
```

Print the predicted value

```
print(predicted_y)
```

```
       1
16.30369
```

> **Printing the fitted line for the simple linear regression**
>
> ### Extract coefficients
>
> ```
> coefficients <- coef(model)
>
> beta_0 <- coefficients[1]
> beta_1 <- coefficients[2]
> # beta_2 <- coefficients[3]
> ```
>
> ### Construct the fitted line expression in LaTeX
>
> ```
> # Construct the fitted line expression in LaTeX, conditionally applying signs
> fitted_line <- paste0("\\hat{y} = ", round(beta_0, 3),
>                       ifelse(beta_1 < 0, " - ", " + ")
>                             , round(abs(beta_1), 3), " \\cdot x_1" #,
>                       # ifelse(beta_2 < 0, " - ", " + ")
>                       #       , round(abs(beta_2), 3), " \\cdot x_2"
>                       )
> ```
>
> ### Print the fitted line
>
> $$\hat{y} = -9.291 + 0.64 \cdot x_1$$

Multiple Regression Using `lm()` Method

Step 1: Load necessary package

```r
# install.packages("dplyr") # Uncomment to install

library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Step 2: Define the data

```r
X1 <- c(0, 1, 2, 3, 4, 5)
X2 <- c(0, 1, 4, 9, 16, 25)
y <- c(9.1, 7.3, 3.2, 4.6, 4.8, 2.9)
```

Step 3: Create a data frame

```r
data <- data.frame(y, X1, X2)
```

```r
print(-9.2907 + (0.6399 * 40))
```

```
[1] 16.3053
```

Step 4: Fit the model

```r
model <- lm(y ~ X1 + X2, data = data)
summary(model)
```

```
Call:
lm(formula = y ~ X1 + X2, data = data)

Residuals:
      1       2       3       4       5       6 
 0.1393  0.5921 -1.8514  0.6086  1.2721 -0.7607 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   8.9607     1.3202   6.788  0.00654 **
X1           -2.5511     1.2418  -2.054  0.13221   
X2            0.2982     0.2384   1.251  0.29964   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.457 on 3 degrees of freedom
Multiple R-squared:  0.7831,	Adjusted R-squared:  0.6385 
F-statistic: 5.416 on 2 and 3 DF,  p-value: 0.101
```

Step 5: Predict y when x1 = 2

```r
new_data <- data.frame(X1 = 2, X2 = 2^2) # X2 = 4
predicted_y <- predict(model, newdata = new_data)
```

Print the predicted value

```r
print(predicted_y)
```

```
       1
5.051429
```

Printing the fitted line for the multiple linear regression

Extract coefficients

```r
coefficients <- coef(model)

beta_0 <- coefficients[1]
beta_1 <- coefficients[2]
beta_2 <- coefficients[3]
```

```r
print(coefficients)
```

```
(Intercept)          X1          X2
  8.9607143  -2.5510714   0.2982143
```

Construct the fitted line expression in LaTeX

```r
# Construct the fitted line expression in LaTeX, conditionally applying signs
fitted_line <- paste0("\\hat{y} = ", round(beta_0, 3),
                      ifelse(beta_1 < 0, " - ", " + ")
                            , round(abs(beta_1), 3), " \\cdot x_1",
                      ifelse(beta_2 < 0, " - ", " + ")
                            , round(abs(beta_2), 3), " \\cdot x_2"
                      )
```

Print the fitted line

$$\hat{y} = 8.961 - 2.551 \cdot x_1 + 0.298 \cdot x_2$$

To calculate each of these matrices in R, follow these steps:

Step 1: Define Matrix X

```
X <- matrix(c(
  1, 0, 0,
  1, 1, 1,
  1, 2, 4,
  1, 3, 9,
  1, 4, 16,
  1, 5, 25
), nrow = 6, ncol = 3, byrow = TRUE)
```

Step 2: Calculate the Transpose of X (X')

```
Xt <- t(X)
Xt
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    1    1    1
[2,]    0    1    2    3    4    5
[3,]    0    1    4    9   16   25
```

Step 3: Calculate X' * X

```
XtX <- Xt %*% X
XtX
```

```
     [,1] [,2] [,3]
[1,]    6   15   55
[2,]   15   55  225
[3,]   55  225  979
```

Step 4: Calculate the Inverse of X' * X

Use solve() to find the inverse.

```
XtX_inv <- solve(XtX)
XtX_inv
```

```
            [,1]       [,2]        [,3]
[1,]  0.82142857 -0.5892857  0.08928571
[2,] -0.58928571  0.7267857 -0.13392857
[3,]  0.08928571 -0.1339286  0.02678571
```

Output Explanation

- `Xt` gives you the transpose of `X`.
- `XtX` shows the result of `X' * X`.
- `XtX_inv` displays the inverse of `X' * X`.

Compute β

$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$,

ε represents the residuals

To estimate β, we minimize the sum of squared residuals by solving:

[$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$]

From above matrix output

```r
# y <- c(9.1, 7.3, 3.2, 4.6, 4.8, 2.9)

y <- t(matrix(c(9.1, 7.3, 3.2, 4.6, 4.8, 2.9), nrow = 1, ncol = 6, byrow = FALSE))

beta = XtX_inv %*% Xt %*% y

print(beta)
```

```
           [,1]
[1,]  8.9607143
[2,] -2.5510714
[3,]  0.2982143
```

We find same results for β

The `lm()` method yields the same results for β_0, β_1 and β_2 as follows:

$$\hat{y} = 8.961 - 2.551 \cdot x_1 + 0.298 \cdot x_2$$

While the [$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$] method as follows:

$$8.961, -2.551, 0.298$$

To answer your question on $X^T y$, I have multiplied `Xt` by `y` see below results:

```
Xt %*% y = [31.9, 61.2, 210.8]
```