Problem Set 2

INF 511

Harii

1 Manual OLS analysis to estimate $\hat{B}$

Conducting OLS analysis on a data set that is simulate.

1.1 Use the following parameters to generate the X covariate matrix.

First, we will need to generate the matrices and vectors that are needed to generate the data. Remember the X matrix is n x p, where n is the number of data observations, and p is the number of parameters. Here, $p = 2$, for the intercept and one slope (i.e., we are only dealing with one co variate, x).

```r
n = 50
p = 2

x0 = rep(1, times = n)


# Randomly draw from a probability distribution to generate
# n values for x1.
x1 =rnorm(n)

# Create the matrix, xmat, using the x0 and x1 column vectors
xmat = cbind(x0, x1)
```

1.2 Create an array of residual error values

Remember that $\epsilon_i \sim N(0, \sigma^2)$. Draw ϵ_i values randomly from a normal distribution.

I took the mean of the residual error values as 0 because it's a common assumption in linear regression modeling that the residuals have a mean of 0. This means that on average, the model predictions will be equal to the observed response values. This assumption can be verified through residual plots and hypothesis tests.

However, this assumption may not always hold in practice, and there are methods to model non-zero mean residuals if necessary. It's important to carefully consider the assumptions being made about the error terms in the model and to check for validity through residual analysis.

```r
# Assign a value for sigma, the residual standard deviation
sigma = sd(x1)
epsilon <- rnorm(n, mean = 0, sd = sqrt(sigma))
```

1.3 Calculate the observed values of Y.

```
# Use the following values of intercept and slope
# betas[1]: intercept
# betas[2]: slope
betas = c(1.50, 1.75)
betas = c(1.50, 1.75)
y = betas[1] + betas[2] * xmat[,2] + epsilon # calculate y using the intercept, ```
```

1.4 Calculate $\hat{B}$

Now we have the data observations. Using the example code already provided, calculate the coefficients that we estimate from the data using OLS, stored in matrix, $\hat{B}$. Use the `solve()` function.

```
# Enter your code here
b_hat = solve(crossprod(xmat), t(xmat) %*% y)
```

2 OLS analysis using `lm()`

Use the `lm()` function to estimate the model coefficients. Store the estimated coefficients in an array.

```
# Enter your code here
# Load the necessary library
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0     v purrr   1.0.1
v tibble  3.1.8     v dplyr   1.1.0
v tidyr   1.3.0     v stringr 1.5.0
v readr   2.1.3     v forcats 1.0.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
data_ <- as.data.frame(xmat)
# Fit the model using the lm() function
fit <- lm(y ~ x0 + x1, data = data_)

# Access the coefficients from the fit object
coefficients <- coef(fit)
```

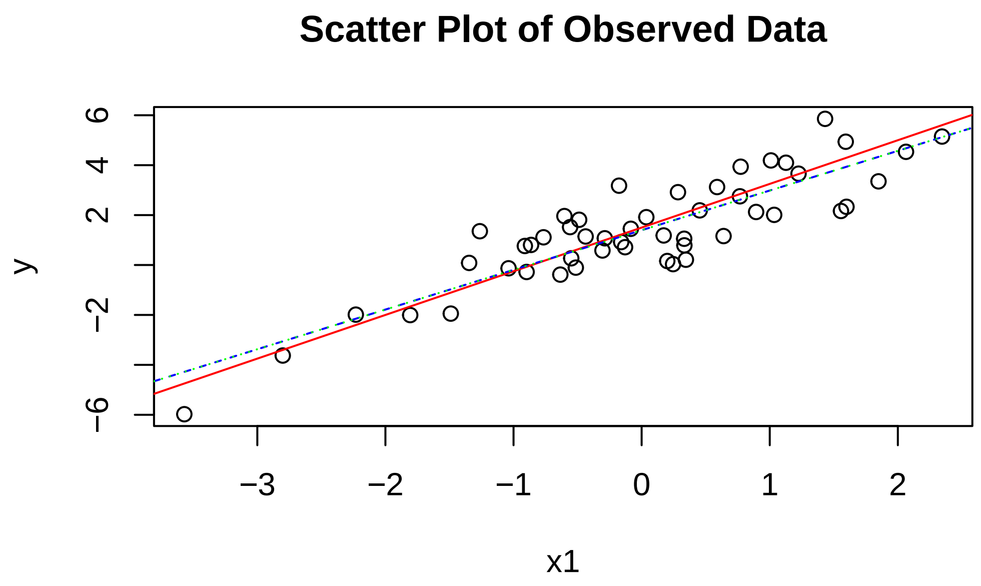
3 Visualize and compare the analyses

Plot the observed data in a scatter plot using `plot()`. As in the code already provided, plot three lines: (1) linear relationship with true values of coefficients, B; (2) linear relationship with coefficients estimated from manual OLS analysis; and (3) linear relationship with coefficients estimated from `lm()`. Each line should be a different color and a different line type (option `lty` in the `abline()` function.) Finally, create chunk options (i.e., using the `#|` syntax) to specify the plot's height and width.

```
# Enter your code here
# Load the necessary library
library(tidyverse)

# Convert the matrix to a data frame
data_ <- as.data.frame(xmat)

# Fit the model using the lm() function
fit <- lm(y ~ x0 + x1, data = data_)

# Access the coefficients from the fit object
coefficients <- coef(fit)

# Plot the observed data
plot(xmat[,2], y, xlab = "x1", ylab = "y", main = "Scatter Plot of Observed Data")

# Add the true values of coefficients, B
abline(a = betas[1], b = betas[2], col = "red", lty = 1)

# Add the coefficients estimated from manual OLS analysis
abline(a = b_hat[1], b = b_hat[2], col = "blue", lty = 2)

# Check for non-finite values in coefficients
any(!is.finite(coefficients))
```

```
[1] TRUE
```

```
  # Remove non-finite values from coefficients
  coefficients <- coefficients[is.finite(coefficients)]

# Add the coefficients estimated from lm()
abline(a = coefficients[1], b = coefficients[2], col = "green", lty = 3)
```



```
# Set the plot height and width
#| plot.height = 6
#| plot.width = 8
```