Assignment 2 - Linear Regression

Code:

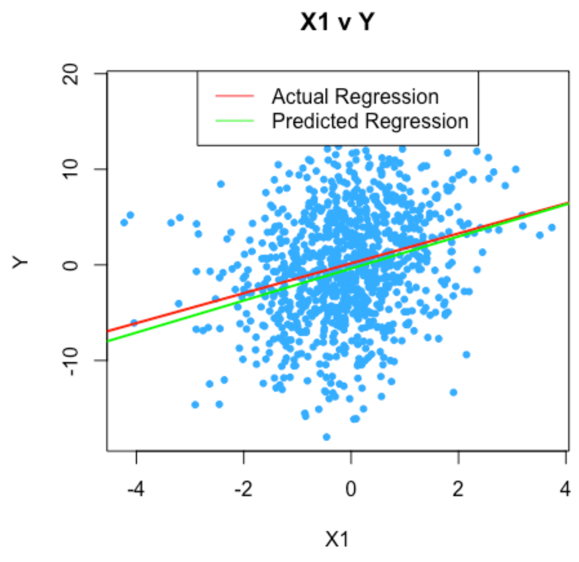
```
linear_regression <- function(X,y) {

 # Add intercept column
 int <- rep(1,length(y))
 X <- cbind(int, X)

 # Solving for Beta
 # Here X is the matrix, t(X) is the transpose of the matrix and y is the vector.
 # We multiply X by its transpose, and find the inverse of the result using solve().
 # We then multiply the result by the transpose of X, and y.
 B <- solve(t(X) %*% X) %*% t(X) %*% y
 B <- t(B)

 # Creating the x matrix for testing data
 X_test <- as.matrix(df_test)

 # Add intercept column to testing data
 int <- rep(1,nrow(X_test))
 X_test <- cbind(int, X_test)

 Y_predicted <- rep(0, nrow(X_test))

 # Predicting Y for the test data
 for (i in 1:nrow(X_test)) {
  for(j in 1:length(B)) {
   Y_predicted[i] <- Y_predicted[i] + (X_test[i,j] * B[j])
  }
 }

 # Taking the common number of rows so we have conformable arrays
 n = min(nrow(df_test),nrow(df_train))

 Y_actual = c(y[0:n])

 # Calculating R squared
 R_sq <- (sum((Y_predicted-mean(Y_actual))^2))/(sum((Y_actual-mean(Y_actual))^2))

 # Printing
 cat("The R squared value is ", R_sq)
 cat("\nThe coefficients are: ", B)
```

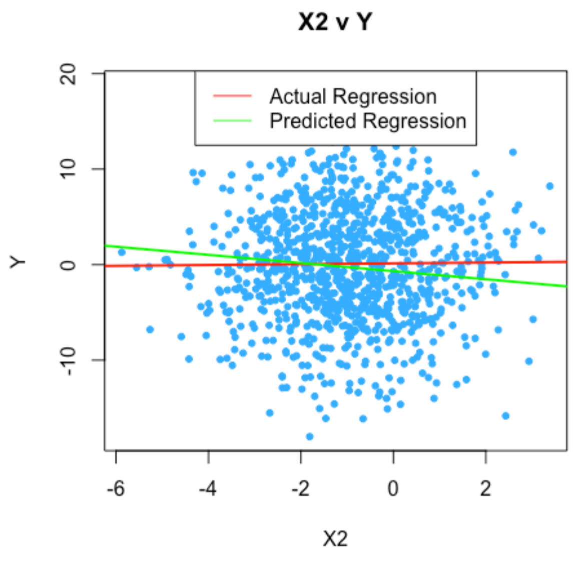
```
  # Comparing results by using lm()
  lm_result <- lm(formula = Y ~ X1 + X2 + X3 + X4 + X5, data = df_train)
  summary(lm_result)

  # Plotting the graphs, showing regression line of training data in red, and predicted line using
the testing data in green
  # For X1
  plot(df_train$X1, df_train$Y, xlab = 'X1', ylab = 'Y', pch = 20, cex = 1, col = "#35b1ff", main =
"X1 v Y")
  abline(lm(df_train$Y ~ df_train$X1), col="red", lwd=2)
  abline(lm(Y_predicted ~ df_test$X1), col="green", lwd=2)
  legend("top", c("Actual Regression", "Predicted Regression"), col=c("red", "green"), lty = 1)

  # For X2
  plot(df_train$X2, df_train$Y, xlab = 'X2', ylab = 'Y', pch = 20, cex = 1, col = "#35b1ff", main =
"X2 v Y")
  abline(lm(df_train$Y ~ df_train$X2), col="red", lwd=2)
  abline(lm(Y_predicted ~ df_test$X2), col="green", lwd=2)
  legend("top", c("Actual Regression", "Predicted Regression"), col=c("red", "green"), lty = 1)

  # For X3
  plot(df_train$X3, df_train$Y, xlab = 'X3', ylab = 'Y', pch = 20, cex = 1, col = "#35b1ff", main =
"X3 v Y")
  abline(lm(df_train$Y ~ df_train$X3), col="red", lwd=2)
  abline(lm(Y_predicted ~ df_test$X3), col="green", lwd=2)
  legend("top", c("Actual Regression", "Predicted Regression"), col=c("red", "green"), lty = 1)

  # For X4
  plot(df_train$X4, df_train$Y, xlab = 'X4', ylab = 'Y', pch = 20, cex = 1, col = "#35b1ff", main =
"X4 v Y")
  abline(lm(df_train$Y ~ df_train$X4), col="red", lwd=2)
  abline(lm(Y_predicted ~ df_test$X4), col="green", lwd=2)
  legend("top", c("Actual Regression", "Predicted Regression"), col=c("red", "green"), lty = 1)

  # For X5
  plot(df_train$X5, df_train$Y, xlab = 'X5', ylab = 'Y', pch = 20, cex = 1, col = "#35b1ff", main =
"X5 v Y")
  abline(lm(df_train$Y ~ df_train$X5), col="red", lwd=2)
  abline(lm(Y_predicted ~ df_test$X5), col="green", lwd=2)
  legend("top", c("Actual Regression", "Predicted Regression"), col=c("red", "green"), lty = 1)

  # Cleanup
  rm(i,j,int,n,y)
```

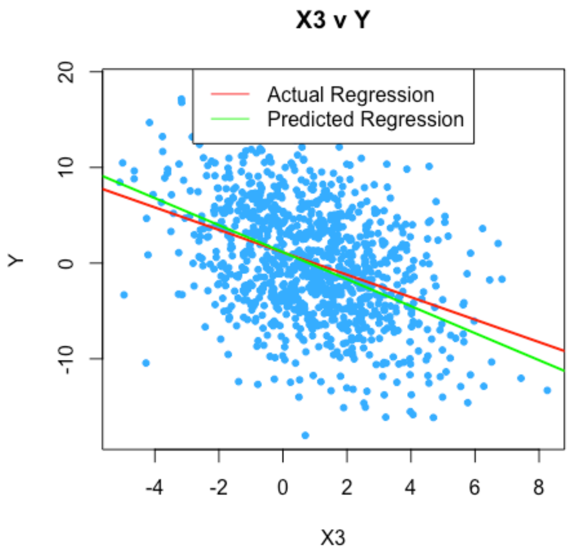
```
}

# Creating dataframes for training and testing data
df_train <- read.csv(file = "TrainData_Group1.csv", header = TRUE, sep = ",")
df_test <- read.csv(file = "TestData_Group1.csv", header = TRUE, sep = ",")

# Creating the y vector (technically a matrix here) for training data
y <- as.matrix(df_train[6])

# Creating the x matrix for training data
X <- as.matrix(df_train[1:5])

# Starting the function
linear_regression(X,y)
```

Output:

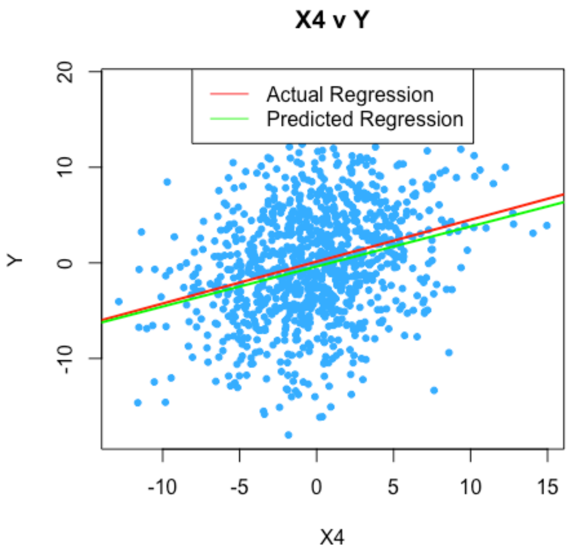
```
The R squared value is  0.9671818
The coefficients are:  2.127945 0.0108312 -0.003680646 -1.305743 0.3976384 2.084183
```



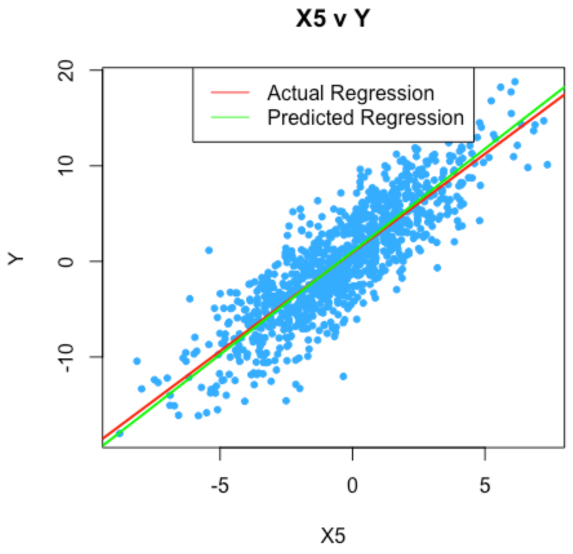




X3 v Y
Actual Regression
Predicted Regression
Y
X3
20
10
0
-10
-4
-2
0
2
4
6
8



X4 v Y
Actual Regression
Predicted Regression
Y
X4
20
10
0
-10
-10
-5
0
5
10
15



X5 v Y
Actual Regression
Predicted Regression
Y
X5
20
10
0
-10
-5
0
5