Statistical Programming with R Assignment 1

Group_6 (Sanjaya Bandara , Irosha sadamali)

1) Linear regression

These are the Packages that is used to build this function

```
library(knitr)
library(tidyverse)
library(ggplot2)
```

Function for Linear regression model

```
# Function
linear_regression <- function(data, dep, indep, intercept = TRUE) {

  #To check input data
  if (!is.data.frame(data)) stop("Data frames must be used as input data.")

  # To check the length and type of dep
   if (!is.character(dep) || length(dep) != 1) {
    stop("A dependent variable (dep) should consist of a character vector with a length of 1.")
  }

  # To check indep is a non-empty and character vector
  if (!is.character(indep) || length(indep) == 0) {
    stop("Independet variables should consist of a character vector with length nonzero.")
  }


  #  The intercept column should be added if specified
  if (intercept) {
    data$intercept <- 1
    indep <- c("intercept", indep)
  }

  # Convert "indep" to a Matrix and name it as X
  X <- as.matrix(data[, indep, drop = FALSE])

  # Response variable(dep)
  y <- data[[dep]]

  # $beta$ Coefficients
  beta <- solve(t(X) %*% X) %*% t(X) %*% y

  # Residuals
  residuals <- y - X %*% beta

  # Degrees of freedom
  df <- nrow(data) - length(beta)

  # Standard errors

  sigma <- sqrt(sum(residuals^2) / (length(y) - ncol(X)))
  se <- sigma * sqrt(diag(solve(t(X) %*% X)))
  # Output of model
  result <- list(
    coefficients = beta,
    se = se,
    dep = dep,
    indep = indep,
    y = y,
    residuals = residuals,
    fitted = X %*% beta,
    sigma = sqrt(sum(residuals^2) / (length(y) - ncol(X)))
  )
  class(result) <- "linear_regression"
  return(result)
}

# Function for getting the Coefficients and Standard error
print.linear_regression <- function(x) {
  cat("       Coefficient     SE\n")
  cat(paste("Intercept   ", round(x$coefficients[1], 4), "   ", round(x$se[1],4), "\n"))
  cat(paste("unemploy    ", round(x$coefficients[2], 4), "   ", round(x$se[2],4), "\n"))
  cat(paste("pop         ", round(x$coefficients[3], 4), "   ", round(x$se[3], 4), "\n"))
  cat(paste("psavert     ", round(x$coefficients[4], 4), "   ", round(x$se[4], 4), "\n"))
}

# Function for getting the Actual and Fitted values with sorted order plot



plot.linear_regression <- function(x) {
  ggplot(data.frame(Actual = x$y, Fitted = x$fitted, Observation = seq_along(x$y),color="actual"),
         aes(x = Observation)) +
    geom_point(aes(y = Actual,color="Actual"), size = 1, alpha = 0.7, shape = 2) +
    geom_line(aes(y = Fitted,color="Fitted"), size = 1) +
    labs(x = "Order", y = "Value", title = "Actual vs Fitted Values(sorted)")

}
```

Finally function is tested by using economics data set

```
my_data <- as.data.frame(economics)
result <- linear_regression(my_data,
                             dep = "pce",
                             indep = c("unemploy", "pop", "psavert"))
```

two methods for linear regression class

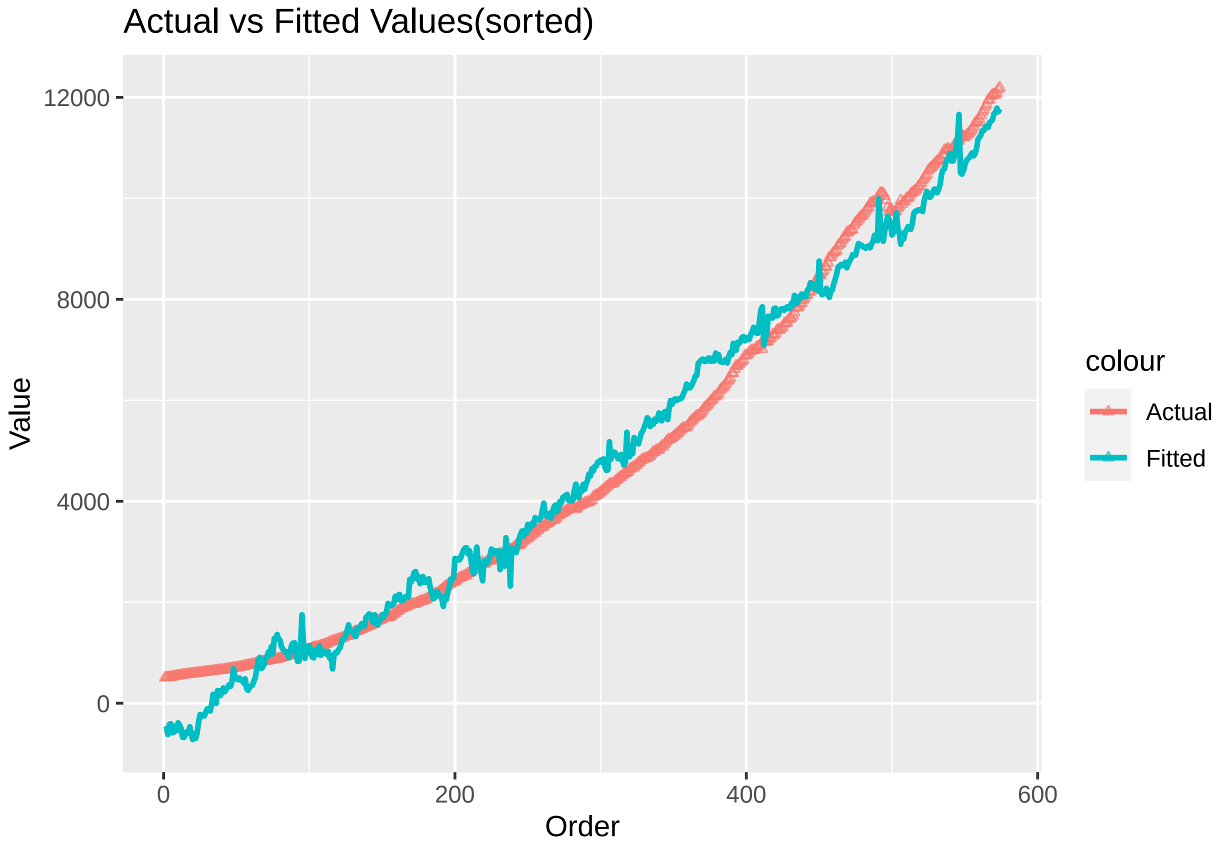
1) Coefficient and Standard Errors of the model

```
print(result)
```

```
       Coefficient     SE
Intercept    -25465.001     434.4301
unemploy     -0.1085     0.0117
pop          0.1143     0.0015
psavert      201.8853     14.7722
```

2) Actual vs Fitted Values(sorted) plot

```
plot(result)
```


Actual vs Fitted Values(sorted)
Value
Order
colour
Actual
Fitted