module\_4\_assignment\_rmd

# Load data  
polynomial\_data <- read.csv('polynomial\_data.txt', header = FALSE, sep = "")  
colnames(polynomial\_data) <- c("X", "Y")  
  
# Create initial variables  
n <- nrow(polynomial\_data)  
X <- polynomial\_data[,1]  
Y <- polynomial\_data[,2]  
  
# Calculate the nth degree for X  
nth\_degree <- function(x, n) {  
 return(x^n)  
}  
  
x2 <- nth\_degree(x = X, n = 2)  
x3 <- nth\_degree(x = X, n = 3)  
x4 <- nth\_degree(x = X, n = 4)  
x5 <- nth\_degree(x = X, n = 5)  
  
# Create the matrices for the nth degree polynomials  
ones <- rep(1, n)  
X3 <- cbind(ones, X, x2, x3)  
X4 <- cbind(ones, X, x2, x3, x4)  
X5 <- cbind(ones, X, x2, x3, x4, x5)  
  
# Calculate the beta hat  
beta\_hat <- function(X, y) {  
 beta\_hat <- solve(t(X) %\*% X) %\*% t(X) %\*% y  
 return(beta\_hat)  
}  
  
beta\_hat\_3 <- beta\_hat(X = X3, y = Y)  
beta\_hat\_4 <- beta\_hat(X = X4, y = Y)  
beta\_hat\_5 <- beta\_hat(X = X5, y = Y)  
  
# Calculate the nth degree y-hat  
y3 <- X3 %\*% beta\_hat\_3  
y4 <- X4 %\*% beta\_hat\_4  
y5 <- X5 %\*% beta\_hat\_5  
  
# Calculate the adjusted R-squared  
adj\_r\_squared <- function(X, betahat, y) {  
 y\_bar <- mean(y)  
 y\_hat <- X %\*% betahat  
  
 SS\_tot <- sum((y - y\_bar)^2)  
 SS\_res <- sum((y - y\_hat)^2)  
   
 n <- length(y)  
 p <- ncol(X) - 1  
 df\_e <- n - p - 1  
 df\_t <- n - 1  
   
 adj\_r\_square <- 1 - ((SS\_res / df\_e) / (SS\_tot / df\_t))  
 return(adj\_r\_square)  
}  
  
adj\_r\_squared\_3 <- adj\_r\_squared(X = X3, betahat = beta\_hat\_3, y = Y)  
adj\_r\_squared\_4 <- adj\_r\_squared(X = X4, betahat = beta\_hat\_4, y = Y)  
adj\_r\_squared\_5 <- adj\_r\_squared(X = X5, betahat = beta\_hat\_5, y = Y)  
  
### Chosen model: n=4  
y\_hat\_4 <- X4 %\*% beta\_hat\_4  
e\_4 <- Y- y\_hat\_4  
mean(e\_4)  
sd(e\_4)  
  
sorted\_residuals <- sort(e\_4)  
e\_quantile <- qnorm((1:n)/n, mean = mean(e\_4), sd = sd(e\_4))  
plot(e\_quantile, sorted\_residuals, main = 'Q-Q Plot',  
 xlab = latex2exp::TeX('$Normal(\\bar{e}, \\sigma\_{e}^2)$'),  
 ylab = latex2exp::TeX('Sorted Residuals'))  
df <- data.frame(x = e\_quantile, y = sorted\_residuals)  
fit <- lm(formula = y~x, data = df[1:(n-1),])  
abline(fit)