midterm\_rmd

### Problem 2 (c)  
# plot density vs theta  
thetas <- seq(0, 1, length.out = 1e4)  
s0 <- function(theta) { return(2 \* (1 - theta)) }  
s1 <- function(theta) { return(2 \* theta) }  
plot(thetas, s0(thetas), type = 'l', col = 'red',  
 main = latex2exp::TeX('$P(\\theta | D) \\;vs.\\;\\theta$'),  
 xlab = latex2exp::TeX('$\\theta$'),  
 ylab = latex2exp::TeX('$P(\\theta | D)$'))  
lines(thetas, s1(thetas), col = 'blue')  
legend("right",  
 legend = c(latex2exp::TeX('$s=0$'),  
 latex2exp::TeX('$s=1$')),  
 col = c('red', 'blue'), lty = c(1,1))  
  
# Load data  
polynomial\_data <- read.csv('xid-94690945\_2.txt', header = FALSE, sep = "")  
colnames(polynomial\_data) <- c("X", "Y")  
  
# Create initial variables  
m <- nrow(polynomial\_data)  
X <- polynomial\_data[,1]  
Y <- polynomial\_data[,2]  
  
# Calculate the nth degree for X  
nth\_degree <- function(x, n) {  
 return(cos(n \* x))  
}  
  
nth <- 10  
polynomial\_list <- lapply(1:nth, function(y) nth\_degree(x=X, n=y))  
  
# Create the matrices for the nth degree polynomials  
halves <- rep(0.5, nth)  
# data\_matrix = polynomial\_list[1:3]   
create\_design\_matrix <- function(halves\_vector=halves, data\_matrix) {  
 design\_mat <- cbind(halves\_vector, do.call(cbind, data\_matrix))  
 return(design\_mat)  
}  
# polynomial\_list[1:2] # list these out in 1:1, 1:2, 1:3, etc.  
polynomial\_degree\_vec <- sapply(seq(1, nth), function(x) c(1,x))  
polynomial\_degree\_groups <- lapply(1:nth, function(x) polynomial\_list[1:x])  
  
# List of the different design matrices for various degree polynomials  
polynomial\_design\_matrix\_list <- lapply(polynomial\_degree\_groups,  
 function(x) create\_design\_matrix(data\_matrix = x))  
  
# Calculate the beta hat  
beta\_hat <- function(X, y) {  
 beta\_hat <- solve(t(X) %\*% X) %\*% t(X) %\*% y  
 return(beta\_hat)  
}  
  
beta\_hat\_list <- lapply(polynomial\_design\_matrix\_list,  
 function(x) beta\_hat(X = x, y = Y))  
  
# 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\_list <- mapply(function(a, b) adj\_r\_squared(X = a, betahat = b, y = Y),  
 polynomial\_design\_matrix\_list, beta\_hat\_list)  
  
par(mfrow = c(1,2))  
plot(1:nth, adj\_r\_squared\_list, type='l',  
 main = latex2exp::TeX('$R\_{Adj}^2 \\; vs. \\; n \\; \\left(for \\;n \\in \\left{1,\\cdots,10 \\right}\\right)$'),  
 xlab = latex2exp::TeX('$n$'), ylab = latex2exp::TeX('$R\_{Adj}^2$'))  
points(1:nth, adj\_r\_squared\_list)  
  
plot(4:nth, adj\_r\_squared\_list[4:nth], type='l', ylim = c(0.8, 0.9),  
 main = latex2exp::TeX('$R\_{Adj}^2 \\; vs. \\; n \\; \\left(for \\;n \\in \\left{4,\\cdots,10 \\right}\\right)$'),  
 xlab = latex2exp::TeX('$n$'), ylab = latex2exp::TeX('$R\_{Adj}^2$'))  
points(4:nth, adj\_r\_squared\_list[4:nth])  
  
### Chosen model: n=5  
y\_hat <- polynomial\_design\_matrix\_list[[5]] %\*% beta\_hat\_list[[5]]  
plot(X,Y, type = 'l',  
 main = 'The Midterm Data Fitted with the n=5 Model')  
legend("topleft", legend = c('Midterm Data', 'Fitted Line'),  
 col = c('black', 'red'), lty = c(1,1))  
lines(X,y\_hat, col = 'red')  
  
e <- Y - y\_hat  
mean(e) # correct  
p <- length(beta\_hat\_list[[5]])  
SSE <- sum(e^2)  
sigma\_hat\_2 <- SSE / (m - p)