module\_3\_assignment\_code

library(latex2exp)  
### 1  
### a  
# sketch P(omega\_1 | x) = [p(x|omega\_1) \* P(omega\_1)] / p(x)  
cauchy <- function(x, a, b) {  
 (1 / (pi \* b)) \* (1 / (1 + ((x - a) / b)^2))  
}  
  
a1 <- 3; a2 <- 2; b <- 5; p\_omega1 <- 0.5; p\_omega2 <- 0.5  
p\_x <- function(x, p\_omega1, p\_omega2, a1, a2, b) {  
 p\_x\_given\_omega1 <- cauchy(x = x, a = a1, b = b)  
 p\_x\_given\_omega2 <- cauchy(x = x, a = a2, b = b)  
   
 prob\_x <- (p\_x\_given\_omega1 \* p\_omega1) +  
 (p\_x\_given\_omega2 \* p\_omega2)  
  
 return(prob\_x)  
}  
  
prob\_x <- p\_x(x = xs,  
 p\_omega1 = p\_omega1, p\_omega2 = p\_omega2,  
 a1 = a1, a2 = a2, b = b)  
  
omega\_given\_x <- function(x, p\_omega1, p\_omega2,  
 a1 = a1, a2 = a2, b = b) {  
 # Calculate p(x)  
 prob\_x <- p\_x(x = x,  
 p\_omega1 = p\_omega1, p\_omega2 = p\_omega2,  
 a1 = a1, a2 = a2, b = b)  
   
 # Calculate p(x|omega\_i)  
 p\_x\_given\_omega <- cauchy(x = x, a = a1, b = b)  
   
 cond\_prob <- (p\_x\_given\_omega \* p\_omega1) / prob\_x  
   
 return(cond\_prob)  
}  
  
xs <- seq(-3e2, 3e2, length.out = 1e3)  
ys <- omega\_given\_x(x = xs,  
 p\_omega1 = p\_omega1, p\_omega2 = p\_omega2,  
 a1 = a1, a2 = a2, b = b)  
  
plot(xs, ys, type = 'l',  
 main = TeX('$P(\\omega\_1 | x) \\; vs.\\; x$'),  
 xlab = TeX('$x$'), ylab = TeX('$P(\\omega\_1 | x)$'))  
### b  
prob\_error <- function(x) {  
 x <- abs(x)  
 0.5 - (1 / pi) \* atan(0.5 \* x)  
}  
  
xs <- seq(-5, 5, length.out = 1e3)  
plot(xs, prob\_error(xs), type = 'l',  
 main = TeX('$P(error) \\; vs. \\; | \\frac{a\_2 - a\_1}{b} | $'),  
 ylab = TeX('$P(error)$'), xlab = TeX('$ | \\frac{a\_2 - a\_1}{b} | $'))