STA 440L - Lab 1

Abbey List

August 30, 2021

```r
# Read data
bikedf <- read.csv("210830_bikecrash.csv")
```

```r
# Using glm function
m1 <- glm(crashes ~ traffic_vol + pct_rural,
          data = bikedf,
          family = "poisson")
round(summary(m1)$coef[,1], 4)
```

```
## (Intercept) traffic_vol   pct_rural
##      5.9822      0.0015     -0.0446
```

```r
# calc.score function, mostly from lab document
calc.score <- function(beta, X, y){

  # Vector to hold sums
  d1 <- rep(0, length(beta))

  # Sum the scoring formula over i
  for(i in 1:length(y)){
    d1 <- d1 + (y[i] - exp(X[i,] %*% beta))[1] * matrix(t(X[i,]))
  }

  return(d1)
}
```

```r
# Calculate Hessian matrix
calc.hess <- function(beta, X, n = 100){

  # Hold sums for Hessian entries
  d1 <- matrix(rep(0, length(beta)*length(beta)), nrow = length(beta))

  # Sum Hessian updates over i
  for(i in 1:n){
    d1 <- d1 + ((exp(X[i,] %*% beta))[1] * matrix(X[i,]) %*% t(matrix(X[i,])))
  }

  # Negate
  d1 = -d1

  return(d1)
}
```

```r
# Create initial beta guess
beta_old <- matrix(c(0.1, 0.1, 0.1), byrow = FALSE)

# Get X and y values from data
Xval <- matrix(c(rep(1, 100), bikedf$traffic_vol, bikedf$pct_rural),
               nrow = 100,
               ncol = 3,
               byrow = FALSE)
yval <- matrix(c(bikedf$crashes), nrow = 1, ncol = 100, byrow = TRUE)

# Initial beta "difference" vector
betadiff <- matrix(c(10,10,10), byrow = FALSE) - beta_old

# Repeat while some betas are still changing by more than 0.00001
while((any(abs(betadiff) > 0.00001))){

  # Update beta
  beta_new <- beta_old - solve(calc.hess(beta_old,
                                         Xval,
                                         100)) %*% calc.score(beta_old,
                                                              Xval,
                                                              yval)

  # Calculate difference between new and old beta
  betadiff <- beta_new - beta_old

  # Replace old beta
  beta_old <- beta_new

  # Check convergence
  if(all(abs(betadiff) < 0.00001)){
    break
  }
}

print(paste("Newton-Raphson (Intercept, traffic_vol, pct_rural):",
            toString(round(beta_new[1], 5)), ",",
            toString(round(beta_new[2], 5)), ",",
            toString(round(beta_new[3], 5))))
```

```
## [1] "Newton-Raphson (Intercept, traffic_vol, pct_rural): 5.98218 , 0.00154 , -0.04456"
```

```r
# Using glm function
m1 <- glm(crashes ~ traffic_vol + pct_rural,
          data = bikedf,
          family = "poisson")
round(summary(m1)$coef[,1], 5)
```

```
## (Intercept) traffic_vol   pct_rural
##     5.98218     0.00154    -0.04456
```

From the output above, the manual Newton-Raphson method provides the same values as the `glm()` function to five decimal places.