Newton-Raphson

Linda Tang

8/30/2021

Output using GLM

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

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

Numerical implementation of Poisson regression

```
# Prepare data for computation
bike$intercept = 1
X <- bike %>% select(intercept, traffic_vol, pct_rural)
y <- bike$crashes

y <- matrix(y, ncol = 1)
X <- data.matrix(X)
colnames(X) <- NULL

# Initialize beta
beta <- c(5, 0.1, 0.1)  # [1] 5.0 0.1 0.1 # current dimension 3*1
```

```
# calculating the score
calc.score <- function(beta, X, y){
  d1 <- rep(0, length(beta))
  for(i in 1:length(y)){
    d1 <- d1 + (y[i] - exp(X[i,] %*% beta)) %*% X[i,]
  }
  return(t(d1)) # returns 3*1 matrix
}

#calculating hessian matrix
calc.hess <- function(beta, X, y){
  d1 <- matrix(rep(0,9), ncol=3)
  for(i in 1:length(y)){
    d1 <- d1 + (exp(X[i,] %*% beta)[1,1]*(X[i,]%*%t(X[i,])))
  }
```

```r
  return(-d1) # returns 3*3 matrix
}
```

```r
iter = 1

while (iter <= 100){ # max 100 iter

  beta_new = beta- t(solve(calc.hess(beta, X, y)) %*% calc.score(beta, X, y))[1,]
  # 1*3 vector = 1*3 vector - t(  3*3 matrix %*% 3*1 matrix    )'s first row
  # 1*3 vector = 1*3 vector - 1*3 vector

  if (dist(rbind(beta, beta_new)) < 0.000000001){
    # using distance between vector to define convergence
    print("satisfied criteria of convergence")
    print("stopped at:")
    print(iter)
    break
  }
  beta <- beta_new
  iter = iter + 1
}
```

```
## [1] "satisfied criteria of convergence"
## [1] "stopped at:"
## [1] 43
```

```r
beta
```

```
## [1]  5.98218054  0.00154064 -0.04455809
```