Solutions: Week 5 Lab

Question 1:

The likelihood function is

$$L(\beta_0, \beta_1|\mathbf{y}, \mathbf{x}) = \prod_{i=1}^{n} \pi^{y_i}(1-\pi)^{1-y_i}$$

The log likelihood function is

$$\begin{aligned} l(\beta_0, \beta_1|\mathbf{y}, \mathbf{x}) &= \sum_{i=1}^{n} y_i \log \pi + \sum_{i=1}^{n}(1-y_i)\log(1-\pi) \\ &= \sum y_i \log \pi + \sum \log(1-\pi) - \sum y_i \log(1-\pi) \\ &= \sum y_i \log \frac{\pi}{1-\pi} + \sum \log(1-\pi) \\ &= \sum y_i(\beta_0 + \beta_1 x_i) - \sum \log\{1 + \exp(\beta_0 + \beta_1 x_i)\} \end{aligned}$$

Taking the derivatives of this function with respect to β_0 and β_1 gives the gradient vector elements.

Question 2:

```
x <- saf[,6]
x <- as.numeric(x)-1
y <- saf[,11]
```

Question 3:

```
llik_log<- function(x,y,beta0,beta1)
{
    sum(y*(beta0+beta1*x)) - sum(log(1+exp((beta0+beta1*x))))
}
```

Question 4:

```
beta0 <- seq(-5,5,len=100)
beta1 <- seq(-5,5,len=100)
llik <- matrix(0,100,100)
for(i in 1:length(beta0))
```

```
    for(j in 1:length(beta1))
    {
        llik[i,j]  <- llik_log(x,y,beta0[i],beta1[j])
    }
contour(beta0,beta1,llik)
image(beta0,beta1,llik)
image(beta0,beta1,llik,col=gray(seq(0.01,0.99,len=50)))
contour(beta0,beta1,llik,add=TRUE,nlevels=50)
persp(beta0,beta1,llik)
```

Question 5:

```
grad_log <- function(x,y,beta0,beta1,scaled=TRUE)
{
    a <- sum(y)   - sum(exp(beta0+beta1*x)/(1+exp(beta0+beta1*x)))
    b <- sum(x*y) - sum((x*exp(beta0+beta1*x))/(1+exp(beta0+beta1*x)))
    if(scaled==TRUE)
    {
        return(c(a,b)/sqrt(a^2+b^2))
    }
    if(scaled==FALSE)
    {
        return(c(a,b))
    }
}
```

Question 6:

```
grad  <- array(dim=c(length(beta0),length(beta1),2))
for(i in 1:length(beta0))
    for(j in 1:length(beta1))
    {
        grad[i,j,] <- grad_log(x,y,beta0[i],beta1[j])
    }
```

Question 7:

Initially we plot 50 level sets of the likelihood surface. Subsequently, we superpose on this graph the scaled gradient vector for a selection of β_0, β_1 configurations.

Question 8:

```
  start0 <- -3.5
  start1 <- -3.2
  Nsteps <- 1000
  alpha  <- 0.01
  beta   <- c(start0,start1)
```

```
contour(beta0,beta1,llik,nlevels=50)
for(i in 1:Nsteps)
{
  beta <- beta + alpha*grad_log(x,y,beta[1],beta[2])
  points(beta[1],beta[2],pch=16)
}
beta
```

By choosing $\alpha = 1$, the taken steps become too big such that they miss the maximum point.