MAS 6024, Assignment 3
Registration Number 170224545

21/01/2018

Part 1

(i). As we know, the linear model is

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

$\beta = (\beta_0, \beta_1)^\top$ is the parameter vector where β_0 is the intercept and β_1 is the gradient. So likelihood for the data set $(\mathbf{X}, \mathbf{y})$:

$$p(\beta; \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} p(\beta; y_i, x_i)$$

When data set likelihood for Normally distributed random variables:

$$p(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp\left(-\frac{(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)}{2\sigma^2}\right)$$

So the log-likelihood is

$$l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = logp(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 - \frac{(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)}{2\sigma^2}$$

As we know, the log-likelihood for the data set is

$$l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = c - \frac{1}{18}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)$$

We can get

$$\sigma^2 = 9$$

$$c = -\frac{n}{2}\log 2\pi - \frac{n}{2}\log\sigma^2 = -\frac{n}{2}\log 18\pi$$

The R code to calculate the value of c is as follow,

```r
data<-data.frame(read.csv('//studata08/home/AC/Act17xs/data.csv'))
# build matrix y
y <- as.matrix(data[2])
# calculate the number of points
n<-length(y)
# calculate the value of c
c<- - n/2*log(18*pi)
c

## [1] -100.8775
```

As we can see here, the value of c is -100.8775.

(ii). We have known in (1) that

$$l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = c - \frac{1}{18}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)$$

Because the value of c is -100.8775,

$$l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = -100.8775 - \frac{1}{18}(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)$$

As we know, $\mathbf{y}$ is the vector of responses,$\mathbf{X}$ is the corresponding design matrix of dimension $n2$, and $\beta = (\beta_0, \beta_1)^\top$ is the parameter vector, we can get

$$(\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta) = \sum_{i=1}^{n}(y_i - \beta_1 x_i - \beta_0)^2$$

$$l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = -100.8775 - \frac{1}{18}\sum_{i=1}^{n}(y_i - \beta_1 x_i - \beta_0)^2$$

The R code to define a function to calculate the log-likelihood is as follow,

```r
# bulid the corresponding design matrix x
one <- c(rep(1,n))
x1 <- data[1]
x0 <- cbind(one, x1)
x <- as.matrix(x0)
# define the function to calculate the log-likelihood
Log_likelihood <- function(y,x,beta_0,beta_1){
  beta <- matrix(c(beta_0,beta_1),nrow = 2, ncol =1)
  value <- -100.8775 -t(y-x%*%beta)%*%(y-x%*%beta) / 18
  return (value)
}
```

(iii). First, we obtain

$$E(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top(\mathbf{y} - \mathbf{X}\beta)$$

The gradient along β_0 and β_1 :

$$\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix}$$

$$\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = -\frac{1}{18} \begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta) = -\frac{1}{18} \begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} E(\beta_0, \beta_1)$$

$$\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} E(\beta_0, \beta_1) = \begin{bmatrix} \sum_{i=1}^n 2(x_i\beta_1 + \beta_0 - y_i) \\ \sum_{i=1}^n 2(x_i^2\beta_1 + \beta_0 x_i - x_i y_i) \end{bmatrix}$$

Now, we want to show that $\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} E(\beta_0, \beta_1) = \frac{\partial}{\partial \beta} E(\beta)$

$$E(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\top (\mathbf{y} - \mathbf{X}\beta)$$

$$E(\beta) = \mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top \mathbf{X}\beta + \beta^\top \mathbf{X}^\top \mathbf{X}\beta$$

$$\frac{\partial}{\partial \beta} E(\beta) = -2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\beta = \begin{bmatrix} \sum_{i=1}^n 2(x_i\beta_1 + \beta_0 - y_i) \\ \sum_{i=1}^n 2(x_i^2\beta_1 + \beta_0 x_i - x_i y_i) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} E(\beta_0, \beta_1)$$

Then,we can get

$$\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = -\frac{1}{18} \begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} E(\beta_0, \beta_1) = -\frac{1}{18} \frac{\partial}{\partial \beta} E(\beta) = -\frac{1}{18} * (-2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\beta)$$

To have the the maximum likelihood, we let $\begin{bmatrix} \frac{\partial}{\partial \beta_1} \\ \frac{\partial}{\partial \beta_0} \end{bmatrix} l(\beta; \mathbf{y}, \mathbf{X}, \sigma^2) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$,

$$-2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\beta = 0$$

$$\beta^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Now, we can determine the β_0 and β_1, the R code is as follow:

```
beta <- solve(t(x)%*%x)%*%t(x)%*%y
beta

##             y
## one 0.8444707
## x   3.6522676
```

we get the β_0 is 0.8444707, the β_1 is 3.6522676, then we substitute these two values into the the function in (2),

```
beta_0 = 0.8444707
beta_1 = 3.6522676
Log_likelihood(y,x,beta_0,beta_1)

##            y
## y -129.3051
```

When the β_0 is 0.8444707,the β_1 is 3.6522676, we get the the maximum likelihood estimates which is -129.3051.

Part 2

(i). First, we consider the vertices of the unit square (i.e. v 1 ,v 2 ,v 3 ,v 4 = (0,0),(1,0),(1,1),(0,1).The starting vertex in this function is X1 which is one of the the vertices. The R code is as follow:

```
proportion <- function(n,p,H,X1){
  v<-0
  # Judge whether the starting vertex is in H or not
  if (list(X1) %in% H){
    v=1}

  a <- X1[[1]]
  b <- X1[[2]]

  for (t in 1:n){
    # sample the type of move
    list0=list(c(0,1),c(1,0),c(1,1))
    c <- sample(x = list0, size = 1, replace = T, prob = c(p/2, p/2, (1-p)))
    a1 <- c[[1]][1]
    b1 <- c[[1]][2]
    a <- a + a1
    if (a != 1){
      a = 0}
    b = b + b1
    if (b != 1){
      b = 0}
    # records the vertices visited at times t
    Xt <- c(a,b)
    # Judge whether the new vertice is in H or not
    if (list(Xt) %in% H){
      v=v+1}

  }
  return(v/n)
}
```

(ii). As we can know from the function to calculate the proportion in (1). the proportion is affected by four elements of n, p, the vertices in the target set H and the starting vertex X1.

We calculate the function by given different n from 1 to 30000 in the case of p = 0, p =0.05, p = 0.3, p= 0.8.We can assume X1 = H = (0,0).The R code is as follow:

```
library(ggplot2)

## Warning:  package 'ggplot2' was built under R version 3.4.3

# how proportions affected by n when p = 0
p <- 0
X1 <- c(0,0)
H <- list(c(0,0))
proportions<-c()
for (n in seq(1,30000,100)){
  proportions <- c(proportions, proportion(n,p,H,X1))
}
n <- seq(1,30000,100)
moving_frame <-as.data.frame(cbind(n,proportions))
ggplot(moving_frame,aes(x = n, y = proportions)) + geom_line()
```

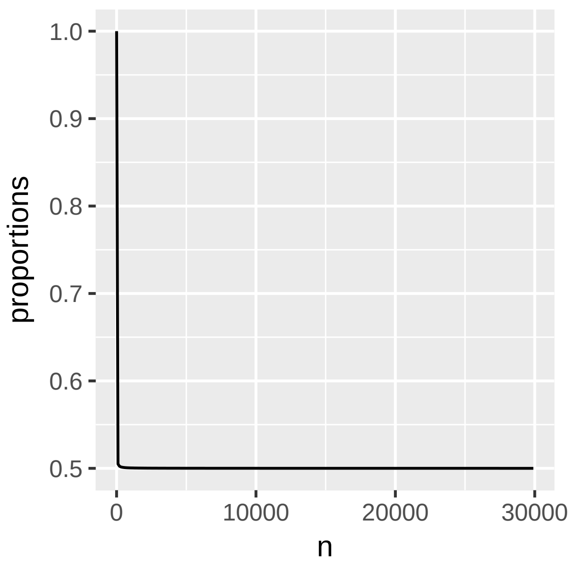

Figure 1: how proportions affected by n when p = 0

```
#how proportions affected by n when p = 0.05
p <- 0.05
X1 <- c(0,0)
H <- list(c(0,0))
proportions<-c()
for (n in seq(1,30000,100)){
  proportions <- c(proportions, proportion(n,p,H,X1))
}
n <- seq(1,30000,100)
moving_frame <-as.data.frame(cbind(n,proportions))
ggplot(moving_frame,aes(x = n, y = proportions)) + geom_line()
```

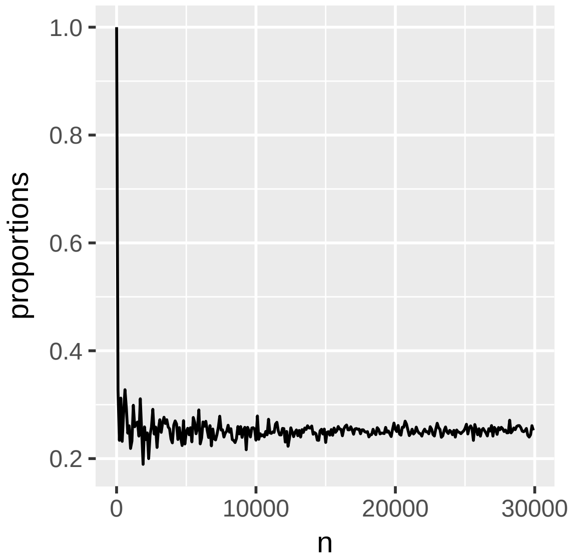

Figure 2: how proportions affected by n when p = 0.05

```
#how proportions affected by n when p = 0.3
p <- 0.3
X1 <- c(0,0)
H <- list(c(0,0))
proportions<-c()
for (n in seq(1,30000,100)){
  proportions <- c(proportions, proportion(n,p,H,X1))
}
n <- seq(1,30000,100)
moving_frame <-as.data.frame(cbind(n,proportions))
ggplot(moving_frame,aes(x = n, y = proportions)) + geom_line()
```

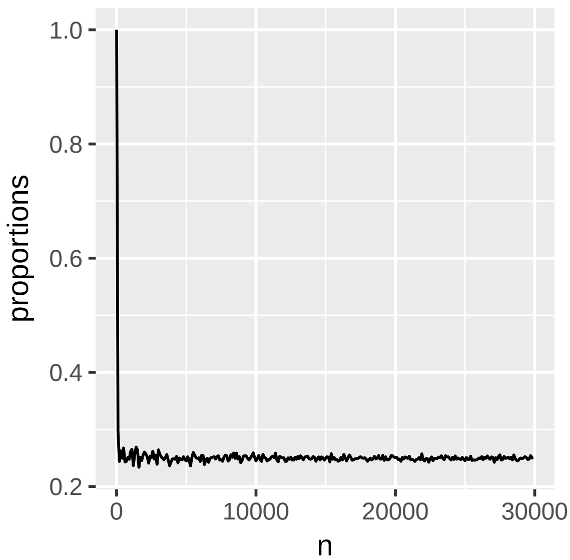

Figure 3: how proportions affected by n when p = 0.3

```r
#how proportions affected by n when p = 0.8
p <- 0.8
X1 <- c(0,0)
H <- list(c(0,0))
proportions<-c()
for (n in seq(1,30000,100)){
  proportions <- c(proportions, proportion(n,p,H,X1))
}
n <- seq(1,30000,100)
moving_frame <-as.data.frame(cbind(n,proportions))
ggplot(moving_frame,aes(x = n, y = proportions)) + geom_line()
```

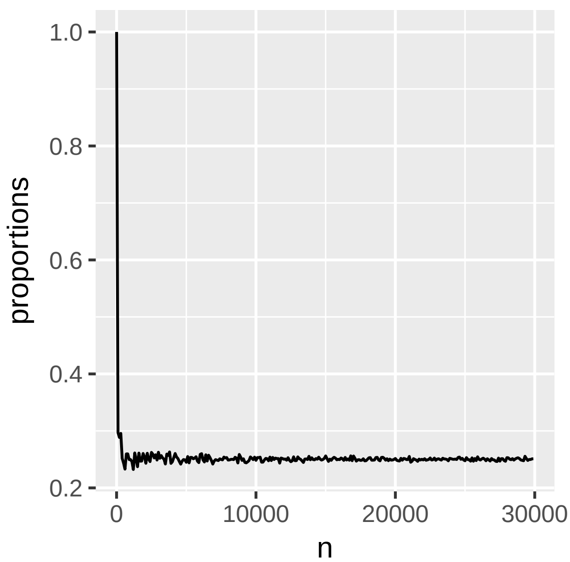

Figure 4: how proportions affected by n when p = 0.8

First we discuss how proportions affected by n. As we can see from those graphs, expect the case of p = 0, the fluctuation of the proportions is converged to 0.25 as we increase the times of the moveing which is n in the function. It means that When n is small, the proportions fluctuate instability and as the n increases, the proportions gradually converge at 0.25.

Then by comparing different graphs with different p value, we can know how proportions affected by p. When p = 0, this is a special case. In the case of p = 0, the object only makes a diagonal move.It means the object only visit the two points on the diagonal.The proportion is 0.5. Expect the case of p = 0, we can see from the graphs that the smaller p value is, the slower the speed of convergence is. It means that when p is small, it need more move times(n) to make the proportions converge to 0.25. When p becomes larger and larger, it need less move times(n) to make proportions converge to 0.25.