Exam 2

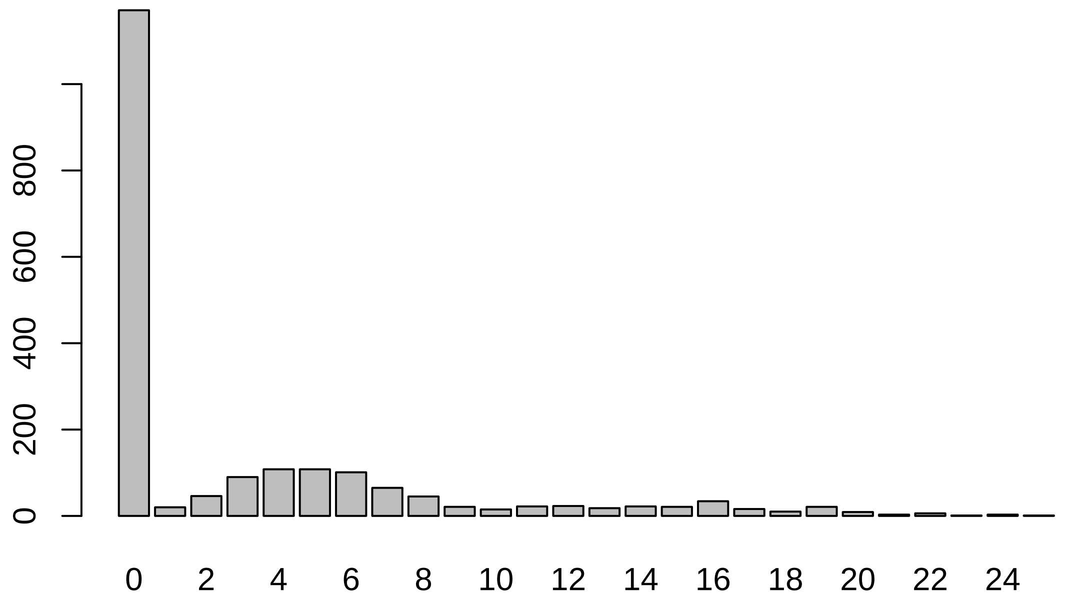
Michael Pena

Spring 2024

Contents

Problem 1

We wanted to investigate marijuana use among college students. We surveyed 2000 randomly selected college students and asked them the following question: "In your best estimation, how many marijuana joints, if any, have you smoked the past thirty days?" We obtained 2000 values ranging from 0 to 27. A barplot of the data is given below.



Let x_i denote the number of joints smoked by the i-th individual surveyed. As shown in the barplot, many students said that they have not smoked at all ($x_i = 0$), and because of this large number of zeros and the fact that the proportions do not monotonically decrease, a Poisson model would not be appropriate. To model the data, we assumed there are three groups: a proportion α of students (group 1) who, for whatever reason, report that they have not smoked marijuana even if this is not true (note that some might not be comfortable revealing that they smoke marijuana). A proportion β of students (group 2) who smoke marijuana occasionally and respond truthfully. For group 2, we assume the number of joints that they smoke follows a Poisson distribution with mean μ. Finally, a proportion $1 - \alpha - \beta$ of students (group 3) who smoke marijuana more often and respond truthfully. For group 3, we assume the number of joints that they smoke follows a Poisson distribution with mean λ. Thus, we assume that each observation x_i comes from the following mixture distribution:

$$f(x_i) = \alpha \; 1_{\{x_i=0\}} + \beta \; \frac{e^{-\mu}\mu^{x_i}}{x_i!} + (1-\alpha-\beta) \; \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$$

where $1_{\{x_i=0\}} = 1$ if $x_i = 0$, and it is zero otherwise.

Your job in this problem is to use the EM algorithm to estimate the parameters $\theta = (\alpha, \beta, \mu, \lambda)$ using the data provided in the dataset `Problem3_Data.csv`.

(a) [3 Points]

Write the (observed data) log-likelihood function.

$$\ell(\theta) = n\Big(log(\beta(1-\alpha-\beta)) - \mu - \lambda\Big) + \sum_{i=1}^{n} \Big(x_i log\big[\frac{\mu\lambda \cdot 1_{\{x_i=0\}}}{(x_i!)^2}\big]\Big)$$

(b) [3 Points]

Describe the complete data, and derive the the complete data log-likelihood.

$$\ell_c(\theta) = \sum_{i=1}^{n} \Big[z_{i1} log\alpha + z_{i2}[log\beta - \mu + x_i log\mu - log|x_i!|] + z_{i3}[log(1-\alpha-\beta) - \lambda + x_i log(\lambda) - log|x_i!|]\Big]$$

(c) [3 points]

Obtain the $Q(\theta', \theta)$ function, including the formulas for the required expectations.

$$Q(\theta'|\theta) = \sum_{i=1}^{N}\sum_{j=1}^{G} E[Z_{ij}]\Big[log[f(x_i|\theta')] + log\pi_j\Big] = \sum_{i=1}^{N}\Big[E[z_{i1}][log\alpha' + log(1_{\{x_i=0\}})] + E[z_{i2}][log\beta' - \mu' + x_i log\mu' - log|x_i!|] + E[z_{i3}][lo$$

where

$$E[z_{i1}] = \frac{\alpha\ 1_{\{x_i=0\}}}{\alpha\ 1_{\{x_i=0\}} + \beta\ \frac{e^{-\mu}\mu^{x_i}}{x_i!} + (1-\alpha-\beta)\ \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}} E[z_{i2}] = \frac{\beta \frac{e^{-\mu}\mu^{x_i}}{x_i!}}{\alpha\ 1_{\{x_i=0\}} + \beta\ \frac{e^{-\mu}\mu^{x_i}}{x_i!} + (1-\alpha-\beta)\ \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}} E[z_{i3}] = \frac{}{\alpha\ 1_{\{x_i=0\}} +$$

(d) [4 points]

Write formulas that maximize $Q(\theta', \theta)$ with respect to θ'. That is, write the update formulas for α, β, μ, and λ.

$$\alpha^* = \frac{\sum_{i=1}^{N} E[z_{i1}]}{N} \beta^* = \frac{\sum_{i=1}^{N} E[z_{i2}]}{N} \mu^* = \frac{\sum_{i=1}^{N} E[z_{i2}]x_i}{\sum_{i=1}^{N} E[z_{i2}]} \lambda^* = \frac{\sum_{i=1}^{N} E[z_{i3}]x_i}{\sum_{i=1}^{N} E[z_{i3}]}$$

(e) [4 points]

Give the formulas for the elements of the gradient of the log-likelihood function.

$$\frac{\partial \ell_c}{\partial \alpha'} = \sum_{i=1}^{n}\left[\frac{E[z_{i1}]}{\alpha'} - \frac{E[z_{i3}]}{1-\alpha'-\beta'}\right] \frac{\partial \ell_c}{\partial \beta'} = \sum_{i=1}^{n}\left[\frac{E[z_{i2}]}{\beta'} - \frac{E[z_{i3}]}{1-\alpha'-\beta'}\right] \frac{\partial \ell_c}{\partial \mu'} = \sum_{i=1}^{n}\left[\frac{x_i}{\mu'} + E[z_{i2}] - 1\right] \frac{\partial \ell_c}{\partial \lambda'} = \sum_{i=1}^{n}\left[\frac{x_i}{\lambda'} + E[z_{i3}] - 1\right]$$

(f) [9 points]

Write an R function to implement the EM algorithm, and a function to compute the gradient of the log-likelihood. Your EM function should input α, β, μ, λ, the data `x`, and a parameter `maxiter` for the maximum number of iteration. To get full credit, you must use the following instructions:

- Write a functions to compute the gradient of the log-likelihood.
- Write a function to compute the required expectations.
- Write a function that implements the EM algorithm. This function should call the gradient and the expectation functions. At each iteration print the iteration number, α, β, $1-\alpha-\beta$, μ, λ, and the norm of the gradient, in that order; use `norm(name_of_gradient, "2)`.
- Show all the functions that you write.
- Use the following parameter values to run your EM function: $\alpha = 1/3$, $\beta = 1/3$, $\mu = 1$, $\lambda = 30$, and `maxiter = 30`.
- Print all iterations.

```r
# making gradient of the llh function
LLgrad <- function(theta,X,EZ){
 # take elements
 a = theta[1]
 b = theta[2]
 mu = theta[3]
 lam = theta[4]
 # input the calculations
dda <- sum(EZ[,1]/a - EZ[,3]/(1-a-b))
ddb <- sum(EZ[,2]/b - EZ[,3]/(1-a-b))
ddmu <- sum(X/mu + EZ[,2] - 1)
ddlam <- sum(X/lam + EZ[,3] - 1)
 # output
 return(c(dda,ddb,ddmu,ddlam))
}

# expectation calculations
expec <- function(F1,F2,F3,P1,P2,P3){
 nu1 = F1 * P1
 nu2 = F2 * P2
 nu3 = F3 * P3
 D = nu1 + nu2 + nu3
 E1 = nu1/D
 E2 = nu2/D
 E3 = nu3/D
 EZ = as.matrix(cbind(E1,E2,E3))
 return(EZ)
}

#  EM algorithm
EM_alg <- function(alpha,beta,mu,lam,X,maxit){
 # define N
 N <- length(X)
 # make prints
 header = paste0("iteration "," alpha     ","    beta      ","    1-beta-alpha","     mu       ","   la
 print(header)

 # loop part
 for(q in 1:maxit){
   PI = c(alpha,beta,1-alpha-beta)
   # get f1
   f1 = rep(0,N)
   for(i in 1:N){
    if(X[i]==0){f1[i]=1}
   }
   # get f2 and f3
   f2 = dpois(X,lambda = mu)
   f3 = dpois(X,lambda = lam)

   # run expectation update
   expec(f1,f2,f3,PI[1],PI[2],PI[3]) -> EZ

   #updates
   alpha1 <- sum(EZ[,1])/N
   beta1 <- sum(EZ[,2])/N
   mu1 <- sum(EZ[,2]*X)/sum(EZ[,2])
   lam1 <- sum(EZ[,3]*X)/sum(EZ[,3])
```

```
    theta1 <- c(alpha1,beta1,mu1,lam1)
    gradnorm <- norm(LLgrad(theta1,X,EZ),"2")

    # print
    print(sprintf("%2.0f         | %8.8f | %8.8f | %8.8f | %8.8f | %8.8f | %8.8f",
                  q,
                  alpha1,
                  beta1,
                  1-alpha1-beta1,
                  mu1,
                  lam1,
                  gradnorm))

    # resets
    alpha1 -> alpha
    beta1 -> beta
    mu1 -> mu
    lam1 -> lam

  }

  #print(ITER)
  print(sprintf("The MLE's are alpha = %f, beta = %f, 1 - beta - alphha = %f, mu = %f, lambda = %f",
                alpha,
                beta,
                1-beta-alpha,
                mu,
                lam))
}
```

```
#run problem
data <- as.matrix(read.csv("Problem1_Data.csv", head =T))
EM_alg(1/3,1/3,1,30,data,30)
```

```
## [1] "iteration  alpha         beta          1-beta-alpha     mu           lambda        gradnorm"
## [1] " 1         | 0.42803480 | 0.44735790 | 0.12460730 | 3.07739304 | 14.55616082 | 1630.91668643"
## [1] " 2         | 0.55859816 | 0.30561955 | 0.13578229 | 4.24722588 | 13.93749323 | 1275.67657399"
## [1] " 3         | 0.58095339 | 0.29161636 | 0.12743025 | 4.70832320 | 14.26251546 | 1299.19975495"
## [1] " 4         | 0.58286093 | 0.29728383 | 0.11985524 | 4.84048783 | 14.61347295 | 1326.50537033"
## [1] " 5         | 0.58314929 | 0.30123410 | 0.11561661 | 4.90009914 | 14.82851860 | 1341.83675089"
## [1] " 6         | 0.58325660 | 0.30334423 | 0.11339917 | 4.93152290 | 14.94324029 | 1349.84994139"
## [1] " 7         | 0.58331098 | 0.30443145 | 0.11225757 | 4.94815032 | 15.00235040 | 1353.96617269"
## [1] " 8         | 0.58333946 | 0.30498745 | 0.11167309 | 4.95681875 | 15.03256017 | 1356.07030084"
## [1] " 9         | 0.58335424 | 0.30527122 | 0.11137454 | 4.96129080 | 15.04797022 | 1357.14414042"
## [1] "10         | 0.58336186 | 0.30541595 | 0.11122219 | 4.96358458 | 15.05582710 | 1357.69182293"
## [1] "11         | 0.58336576 | 0.30548975 | 0.11114450 | 4.96475753 | 15.05983241 | 1357.97107392"
## [1] "12         | 0.58336775 | 0.30552737 | 0.11110488 | 4.96535640 | 15.06187415 | 1358.11343838"
## [1] "13         | 0.58336877 | 0.30554655 | 0.11108468 | 4.96566192 | 15.06291492 | 1358.18601210"
## [1] "14         | 0.58336929 | 0.30555632 | 0.11107439 | 4.96581772 | 15.06344544 | 1358.22300705"
## [1] "15         | 0.58336955 | 0.30556131 | 0.11106914 | 4.96589715 | 15.06371587 | 1358.24186517"
## [1] "16         | 0.58336969 | 0.30556385 | 0.11106647 | 4.96593765 | 15.06385372 | 1358.25147797"
## [1] "17         | 0.58336976 | 0.30556514 | 0.11106510 | 4.96595829 | 15.06392399 | 1358.25637802"
## [1] "18         | 0.58336979 | 0.30556580 | 0.11106441 | 4.96596881 | 15.06395981 | 1358.25887577"
## [1] "19         | 0.58336981 | 0.30556614 | 0.11106405 | 4.96597418 | 15.06397807 | 1358.26014898"
## [1] "20         | 0.58336982 | 0.30556631 | 0.11106387 | 4.96597691 | 15.06398737 | 1358.26079798"
## [1] "21         | 0.58336982 | 0.30556640 | 0.11106378 | 4.96597830 | 15.06399212 | 1358.26112880"
## [1] "22         | 0.58336983 | 0.30556644 | 0.11106373 | 4.96597901 | 15.06399454 | 1358.26129744"
## [1] "23         | 0.58336983 | 0.30556646 | 0.11106371 | 4.96597938 | 15.06399577 | 1358.26138340"
## [1] "24         | 0.58336983 | 0.30556648 | 0.11106370 | 4.96597956 | 15.06399640 | 1358.26142721"
## [1] "25         | 0.58336983 | 0.30556648 | 0.11106369 | 4.96597966 | 15.06399672 | 1358.26144955"
## [1] "26         | 0.58336983 | 0.30556648 | 0.11106369 | 4.96597970 | 15.06399688 | 1358.26146093"
## [1] "27         | 0.58336983 | 0.30556649 | 0.11106369 | 4.96597973 | 15.06399696 | 1358.26146674"
## [1] "28         | 0.58336983 | 0.30556649 | 0.11106368 | 4.96597974 | 15.06399701 | 1358.26146969"
## [1] "29         | 0.58336983 | 0.30556649 | 0.11106368 | 4.96597975 | 15.06399703 | 1358.26147120"
## [1] "30         | 0.58336983 | 0.30556649 | 0.11106368 | 4.96597975 | 15.06399704 | 1358.26147197"
## [1] "The MLE's are alpha = 0.583370, beta = 0.305566, 1 - beta - alphha = 0.111064, mu = 4.965980, la
```

(g) [3 Points]

Using your parameter estimates, explain your findings. Your explanation should involve interpretation of all of the parameters in the context of the problem.

$$f(\theta|x_i) = 0.58337 \cdot 1_{\{x_i=0\}} + 0.305566\frac{e^{4.965980}4.965980^{x_i}}{x_i!} + 0.111064\frac{e^{15.063997}15.063997^{x_i}}{x_i!}$$

0.583370 is the proportion of people in group 1 who mislead about their non-use. 0.305566 is the proportion of people in group 2 or have moderate use. 0.111064 is the proportion of people in group 3 or have frequent use.

4.965980 is the average of the amount of joints per month for people in group 2 15.063997 is the average of amount of joints per month for the people in group 3

(h) [2 points]

We met a student who said that she smoked 9 joints in the past 30 days. What is the probability that this student belongs to each of the three groups, 1, 2, or 3? Which group would you classify this student to? Heuristic explanations don't get any points!

```r
f1 <- 0
f2 <- dpois(9,lambda = 4.965980)
f3 <- dpois(9,lambda = 15.063997)
b <- 0.305566
g <- 0.111064

EZ2 <- f2*b/(f2*b + f3*g)
EZ3 <- f3*g/(f2*b + f3*g)

EZ2;EZ3
```

```
## [1] 0.7545084
```

```
## [1] 0.2454916
```

Probability that she will be in group 1 is 0 because she admitted to actually smoking

Probability that she will be in group is 2 very likely at about 0.7545084 while the prob. that she will be in the last group is 0.2454916 which is less likely.

I want to classify her in group 2 as the expected value for that group is 4.965980 and 9 is very closer to this number than it is 15.1.

(i) [5 points]

Write an R function to numerically approximate the observed information matrix, and use it to obtain the standard errors of the parameter estimates.

```r
INFOMAT <- function(a,b,M,L,EZ,X){
h <- 10^(-7)

# form matrix

H <- matrix(0,4,4)

#element wise
H[1,1] = sum(EZ[,1]/(a*h) - EZ[,3]/(1 - a*h - b)) - sum(EZ[,1]/(a) - EZ[,3]/(1 - a - b))
H[2,2] = sum(EZ[,2]/(b*h) - EZ[,3]/(1 - a - b*h)) - sum(EZ[,2]/(a) - EZ[,3]/(1 - a - b))
H[3,3] = sum(X/(M*h)+EZ[,2]-1) - sum(X/(M)+EZ[,2]-1)
H[4,4] = sum(X/(L*h)+EZ[,3]-1) - sum(X/(L)+EZ[,3]-1)

H[1,2] = sum(EZ[,1]/(a) - EZ[,3]/(1 - a - b*h)) - sum(EZ[,1]/(a) - EZ[,3]/(1 - a - b))
H[2,1] <- H[1,2]

(1/h)*H -> I

return(I)
}


   PI = c(0.583370,0.305566,0.111064)
   # get f1
   f1 <- as.numeric(data == 0)
   # get f2 and f3
   f2 = dpois(data,lambda = 4.965980)
   f3 = dpois(data,lambda = 15.063997)

   # run expectation update
   expec(f1,f2,f3,PI[1],PI[2],PI[3]) -> EZ

# get Infomat
I <- INFOMAT(0.583370,0.305566,4.965980,15.063997,EZ,data)

# get diagnaols
1/diag(I) -> Vs
length(data) -> N
# these are the standard errors
TSE <- sqrt(Vs/N)

sprintf("standard error for alpha is %1.6e, for beta is %1.6e, for mu is %1.6e, for lambda is %1.6e",
        TSE[1],
        TSE[2],
        TSE[3],
        TSE[4])
```

```
## [1] "standard error for alpha is 5.000001e-11, for beta is 4.999996e-11, for mu is 6.237968e-11, for
```

Problem 2

Consider the function

$$g(x) = 30x^2(1-x)^2, \quad 0 < x < 1.$$

(a) [4 points]

Write an R function that uses the basic Monte Carlo method to compute the integral

$$\theta = \int_a^b g(x)dx.$$

for arbitrary values a and b that range in the interval [0,1], and provides a $100(1-\alpha)\%$ confidence interval for θ. Specifically,

- the input to your program must be a, b, number of simulated values, a seed for the random number generation, and a confidence level CL with default value of 0.95. Your program must check that the input variables $0 \le a < b \le 1$, and the confidence level CL is between 0 and 1. If not, the program must stop and print an error message.
- your program must generate values from uniform(a,b) for the Monte Carlo estimation.
- your program must output and print the Monte Carlo estimate of θ, its estimate of standard error, and a $100(1-\alpha)\%$ confidence interval for θ.
- After writing the program, use it to obtain the value of the integral for $a = 0.2$ and $b = 0.6$. Use 10,000 simulated values and seed 338. Your output should include the Monte Carlo estimate of θ, its estimate of standard error, and a 95% confidence interval for θ.

```r
MC <- function(a,b,N,see.d,CL=.95){
  set.seed(see.d)
  # case one
    if( 0 <= a && a < b && b <= 1 && CL >= 0 && CL <= 1){
      #generate from unif
      u <- runif(N,a,b)
      #take monte carlo
      g.u = 30*u^2 * (1-u)^2
      theta.bar = mean(g.u)
      theta = (b-a)*theta.bar
      theta.se = (b-a)*sd(g.u)/sqrt(N)
      #conf interval
      zSE = qnorm(CL)*theta.se
      CI.vals = c(theta - zSE, theta + zSE)
      CI = c(min(CI.vals),max(CI.vals))
      #outputs
      return(list(theta = theta, "Stand Err" = theta.se, "Conf Interval" = CI))
    }
  # case two: invalid parameters
    else{
      print("Error!: Invalid parameters")
    }
}

MC(.2,.6,10000,338)
```

```
## $theta
## [1] 0.6245553
```

```
##
## $`Stand Err`
## [1] 0.001337473
##
## $`Conf Interval`
## [1] 0.6223554 0.6267553
```

(b) [3 points]

Repeat part (a), but instead of generating values from uniform(a,b), suppose that you generate values from the Beta distribution with parameters $\alpha = 2$ and $\beta = 2$. Explain how you can use the beta distribution to estimate the integral. In your explanation specify what expectation you will be estimating and how you will estimate it. No R code is needed here.

(c) [2 points]

Write a formula for the variance of the estimate of θ in part (b), as a function of $Y \sim Beta(\alpha = 2, \beta = 2)$ and explain how you would estimate this variance in you R program. Do not write R code here. Only give an explanation of what R function you would use and how.

$$Var(\hat{\theta}) = \hat{\theta} - \hat{\theta}^2 = \frac{1}{N}\sum_{i=1}^{N}\frac{g(a+(b-a)y_i)(b-a)}{f(y_i)} - \left[\frac{1}{N}\sum_{i=1}^{N}\frac{g(a+(b-a)y_i)(b-a)}{f(y_i)}\right]^2$$

This would be easy to impliment into the code because after we find theta-hat, we could calculate variance as above; this is useful for calculating Standard Error with is simply $\sqrt{var(\hat{\theta})/N}$.

(d) [4 points]

Write an R code to implement the methods that you described in parts (b) and (c). Your program should be written for general a, b, and CL values again, and output and print the Monte Carlo estimate of θ, its estimate of standard error, and a 95% confidence interval for θ. Run your program using the same values for $a = 0.2$, $b = 0.6$, 10,000 simulated values, seed 338, and confidence level 95%. you must use the `rbeta` function to generate values from the beta distribution.

```
MC2 <- function(a,b,N,see.d,CL=.95){
 y <- rbeta(N,2,2)
  set.seed(see.d)
  # case one
    if( 0 <= a && a < b && b <= 1 && CL >= 0 && CL <= 1){
    # pass y through beta(2,2)
    fy = 6*y*(1-y)
    # pass through g(x)
    v = a+(b-a)*y
    gy = 30*v^2*(1-v)^2
  # theta
    theta = mean(gy*(b-a)/fy)
    thetaVAR = theta - theta^2
    thetaSE = sqrt((thetaVAR)/N)


    zSE = qnorm(CL)*thetaSE
    CI.vals = c(theta - zSE, theta + zSE)
    CI = c(min(CI.vals),max(CI.vals))
 #outputs
```

```
    return(list(theta = theta, "Stand Err" = thetaSE, "Conf Interval" = CI))
}
  # case two: invalid parameters
    else{
      print("Error!: Invalid parameters")
    }
}
```

```
# run problem
MC2(.2,.6,10000,338)
```

```
## $theta
## [1] 0.6331223
##
## $`Stand Err`
## [1] 0.004819527
##
## $`Conf Interval`
## [1] 0.6251949 0.6410498
```

Problem 3

Consider the function

$$h(x) = \frac{x^3}{1+x^2}.$$

Let X be a random variable with density function

$$f(x) = \frac{e^{-x}}{1-e^{-1}} \quad 0 \le x \le 1.$$

We want to estimate $E(h(X)$, the expected value of $h(X)$.

(a) [3 points]

Write an R function that uses the basic Monte Carlo method to approximate the required integral. Specifically, generate 100,000 values from the random variable X, and use these values to estimate $E(h(X))$. Your function should output the Monte Carlo estimate of $E(h(X))$ and its estimate of standard error. Use seed 666.

```
# setting seed
set.seed(666)
#generate random values
n = 100000
u <- runif(n)
fu <- exp(-u)*u^3/((1 - exp(-1))*(1+u^2))
#expectation
E <- mean(fu)
# standard error
SE <- sd(fu)/sqrt(100000)
E
```

```
## [1] 0.1141116
```

```
SE
```

```
## [1] 0.0003046412
```

(b) [3 points]

Here you repeat part (a), but use the method of Antithetic Sampling to estimate $E(h(X)] and its standard error. Write a step by step algorithm (pseudo code) for solving this problem. No R code here. [Hint: you need to generate 50,000 samples from Uniform(0,1).]

- save 50000 samples from a uniform distribution (0,1) in a vector u
- render the first part of the function were `fu1 = exp(-u)*u^3/((1 - exp(-1))*(1+u^2))`
- render second part by making v = 1 - u and then set `fu2 = exp(-v)*v^3/((1 - exp(-1))*(1+v^2))`
- theta will be the mean of the sum of vectors fu1 and fu2
- StandErr will be the `sqrt((1/(2*n)))*(sd(fu1+fu2))`

(c) [4 points]

Write an R function to implement the algorithm. Your function should output the estimate of $E(h(X))$ and its estimate of standard error. Use seed 666.

```r
# set seed
set.seed(666)
# render samples
u <- runif(n/2)
v = 1 - u
# render discrete functions
fu1 = exp(-u)*u^3/((1 - exp(-1))*(1+u^2))
fu2 = exp(-v)*v^3/((1 - exp(-1))*(1+v^2))
# get theta and SE
thetaAnti = (1/n)*sum(fu1 + fu2)
thetaAntiSE = sqrt((1/(2*n)))*(sd(fu1+fu2))
# print
thetaAnti;thetaAntiSE
```

```
## [1] 0.1137378
```

```
## [1] 6.870699e-05
```

(d) [3 points]

Explain the relationship between the standard error of the estimate of $E(h(X))$ obtained using the method of Antithetic Sampling as compared to the standard error of the estimate of $E(h(X))$ obtained using the basic Monte Carlo method. Show this relationship by computing relevant quantities using your R code.

The antithetical method deals with two functions that are negatively correlated; this method should render a lower S.E.

```r
real = 0.113762652609
abs(real-0.1141116)
```

```
## [1] 0.0003489474
```

```r
abs(real-0.1137378)
```

```
## [1] 2.485261e-05
```