Question 1

Part A

(a), (b), (e), and (f) will converge to A as n goes to infinity. Because they are first density and second the support of f is contained in the support of h.

(c) is not because it is not the we want to use for importance sampling as it doesn't have the indicator function like in A.

(d) is not because the support of f does not contain in the support of h

Part B

Setup functions:

```
unif<-runif(1000,0,1)
h_x<-function(x){
  abs(cos(x+x^2)*exp(-2*x))
}
se_calculator<-function(h,f,est,n){
  var<-sum((h/f-est)**2)/(n-1)
  s<-var**0.5
  s/(n)**0.5
}
n=1000
```

Function (a):

```
inv_f_a<- function(x){
  -log(1-x)
}
f_a<-function(x){
  exp(-x)
}
f_a_x<-inv_f_a(unif)

f_a_y<-f_a(f_a_x)
h_a_y<-h_x(f_a_x)
mean_a<-sum(h_a_y/f_a_y)/n
se_a<-se_calculator(h_a_y,f_a_y,mean_a,n)
se_a/mean_a # useful
mean_a+qnorm(0.05)*se_a
mean_a+qnorm(0.95)*se_a
```

```
> mean_a
[1] 0.373683
```

Confidence Interval:

```
> mean_a+qnorm(0.05)*se_a
[1] 0.3569571
> mean_a+qnorm(0.95)*se_a
[1] 0.390409
```

The 90% confidence interval is about [0.357, 0.390]

Meaningful?

Yes, because the relative error is smaller than 1.

```
> se_a/mean_a # useful
[1] 0.02721194
```

Function (b):

```
inv_f_b<- function(x){
  -1/2*log(1-x)
}
f_b<-function(x){
  2*exp(-2*x)
}

f_b_x<-inv_f_b(unif)
f_b_y<-f_b(f_b_x)
h_b_y<-h_x(f_b_x)
mean_b<-sum(h_b_y/f_b_y)/1000
se_b<-se_calculator(h_b_y,f_b_y,mean_b,n)
se_b/mean_b # useful
mean_b+qnorm(0.05)*se_b
mean_b+qnorm(0.95)*se_b
```

```
> mean_b
[1] 0.3870542
```

Confidence Interval:

```
> mean_b+qnorm(0.05)*se_b
[1] 0.3799592
> mean_b+qnorm(0.95)*se_b
[1] 0.3941492
```

The 90% confidence interval is about [0.380, 0.394]

Meaningful?

Yes, because the relative error is smaller than 1.

```
> se_b/mean_b # useful
[1] 0.01114431
```

Function (e):

```
inv_f_e<- function(x){
  -1/3*log(1-x)
}
f_e<-function(x){
  3*exp(-3*x)
}
f_e_x<-inv_f_e(unif)
f_e_y<-f_e(f_e_x)
h_e_y<-h_x(f_e_x)
mean_e<-sum(h_e_y/f_e_y)/1000
se_e<-se_calculator(h_e_y,f_e_y,mean_e,n)
se_e/mean_e # useful
mean_e+qnorm(0.05)*se_e
mean_e+qnorm(0.95)*se_e
```

```
> mean_e
[1] 0.3938754
```

Confidence Interval:

```
> mean_e+qnorm(0.05)*se_e
[1] 0.3813041
> mean_e+qnorm(0.95)*se_e
[1] 0.4064467
```

The 90% confidence interval is about [0.381, 0.406]

Meaningful?

Yes, because the relative error is smaller than 1.

```
> se_e/mean_e # useful
[1] 0.01940413
```

Function (f):

```
u_1<-runif(1000)
u_2<-runif(1000)
x1_box_muller<-abs(sqrt(-2*log(u_1))*cos(2*pi*u_2))
f_f<-function(x){
  2/sqrt(2*pi)*exp((-x^2)/2)
}
mean_f<-sum(h_x(x1_box_muller)/f_f(x1_box_muller))/1000
se_f<-se_calculator(h_x(x1_box_muller),f_f(x1_box_muller),mean_f,n)
se_f/mean_f # useful
mean_f+qnorm(0.05)*se_f
mean_f+qnorm(0.95)*se_f
```

```
> mean_f
[1] 0.3716307
```

Confidence Interval:

```
> mean_f+qnorm(0.05)*se_f
[1] 0.3536431
> mean_f+qnorm(0.95)*se_f
[1] 0.3896184
```

The 90% confidence interval is about [0.354, 0.390]

Meaningful?

Yes, because the relative error is smaller than 1.

```
> se_f/mean_f # useful
[1] 0.02942627
```

Question 2

Part A

(a) $U_1, \ldots, U_n$ i.i.d. samples from Uniform($[-1, 1]^3$). Then the law of large numbers implies that

$$\frac{1}{n}\sum_{i=1}^{n} h(U_i) \approx I.$$

(b) if we wanted to estimate the volume of a region A in $[0, 1]^d$, then we could consider h to be the indicator function of A. That is, we could estimate A's volume with:

$$\frac{\#\{U_i \in A\}}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_A(U_i) \approx \int_{[0,1]^d} \mathbf{1}_A(x)dx = \mathrm{Vol}(A).$$

For the quantity **(1) and (2)** we can use **(a)** to estimate

```
n = 1000
x1<-runif(n,-1,1)
x2<-runif(n,-1,1)
x3<-runif(n,-1,1)
x<-x1^2+x2^2+x3^2
q1<-8*sum(h_muti(x1,x2,x3))/n
```

```
> q1
[1] 2.673399
```

```
x2_1<-as.numeric(x1<=1 & x1>=0)
x2_2<-as.numeric(x2<=1 & x2>=0)
x2_3<-as.numeric(x3<=1 & x3>=0)
q2<-8*sum(x2_1*x2_2*x2_3*h_muti(x1,x2,x3))/n
```

```
> q2
[1] 0.2529547
```

For the quantity **(4)** we can use **(b)** to estimate

```
8*sum(as.numeric(x<=1.1)*x2_1*x2_2*x2_3)/n
```

```
> 8*sum(as.numeric(x<=1.1)*x2_1*x2_2*x2_3)/n
[1] 0.52
```

We cannot use this uniform sample to estimate **(3)** because **(3)** is covering the real line R, but our sample is only covering Uniform($[-1, 1]^3$).

Part B

Setup functions:

```
b =1/sqrt(2)
n <-1000
x1_sample<-rnorm(n, mean = 0, sd = b)
x2_sample<-rnorm(n, mean = 0, sd = b)
x3_sample<-rnorm(n, mean = 0, sd = b)

f_muti<-function(x1,x2,x3,b){
  (2*pi*b**2)^(-3/2)*exp(-(x1^2+x2^2+x3^2)/(2*b**2))
}
h_muti<-function(x1,x2,x3){
  cos(x1+x2)*exp(-(x1^2+x2^2+x3^2))
}
se_calculator<-function(h,f,est,n){
  var<-sum((h/f-est)**2)/(n-1)
  s<-var**0.5
  s/(n)**0.5
}
h_y<-h_muti(x1_sample,x2_sample,x3_sample)
f_y<-f_muti(x1_sample,x2_sample,x3_sample,b)
```

Quantity(a)

```
indicator_i1<-x1_sample^2+x2_sample^2+x3_sample^2
in1<-as.numeric(indicator_i1<=1)
estimator1<-sum((in1*h_y/f_y))/1000
estimator1+qnorm(0.05)*se_calculator(in1*h_y,f_y,estimator1,n)
estimator1+qnorm(0.95)*se_calculator(in1*h_y,f_y,estimator1,n)
```

```
> estimator1
[1] 2.032979
> estimator1+qnorm(0.05)*se_calculator(in1*h_y,f_y,estimator1,n)
[1] 1.908116
> estimator1+qnorm(0.95)*se_calculator(in1*h_y,f_y,estimator1,n)
[1] 2.157841
```

The 90% confidence interval is about [1.908, 2.158]

Quantity(b)

```
in2_1<-as.numeric(x1_sample<=1 & x1_sample>=0)
in2_2<-as.numeric(x2_sample<=1 & x2_sample>=0)
in2_3<-as.numeric(x3_sample<=1 & x3_sample>=0)
in2<-in2_1*in2_2*in2_3
estimator2<-sum((in2*h_y/f_y))/1000
se_calculator(in2*h_y,f_y,estimator2,n)
estimator2+qnorm(0.05)*se_calculator(in2*h_y,f_y,estimator2,n)
estimator2+qnorm(0.95)*se_calculator(in2*h_y,f_y,estimator2,n)
```

```
> estimator2
[1] 0.2710625
> estimator2+qnorm(0.05)*se_calculator(in2*h_y,f_y,estimator2,n)
[1] 0.2175746
> estimator2+qnorm(0.95)*se_calculator(in2*h_y,f_y,estimator2,n)
[1] 0.3245504
```

The 90% confidence interval is about [0.218, 0.325]

Quantity(c)

```
estimator3<-sum((h_y/f_y))/1000
estimator3+qnorm(0.05)*se_calculator(h_y,f_y,estimator3,n)
estimator3+qnorm(0.95)*se_calculator(h_y,f_y,estimator3,n)
```

```
> estimator3
[1] 3.371076
> estimator3+qnorm(0.05)*se_calculator(h_y,f_y,estimator3,n)
[1] 3.240953
> estimator3+qnorm(0.95)*se_calculator(h_y,f_y,estimator3,n)
[1] 3.501199
```

The 90% confidence interval is about [3.241, 3.501]

Quantity(d)

```
in4<-as.numeric(indicator_i1<=1.1)
estimator4<-sum((in4*in2/f_y))/1000
estimator4+qnorm(0.05)*se_calculator(in4*in2,f_y,estimator4,n)
estimator4+qnorm(0.95)*se_calculator(in4*in2,f_y,estimator4,n)
```

```
> estimator4
[1] 0.4537726
> estimator4+qnorm(0.05)*se_calculator(in4*in2,f_y,estimator4,n)
[1] 0.3206582
> estimator4+qnorm(0.95)*se_calculator(in4*in2,f_y,estimator4,n)
[1] 0.5868871
```

The 90% confidence interval is about [0.321, 0.587]

Question 3

Part A

(a)

let $a_1 = \mathbb{P}(X \in [a,b])$ Such $X \sim bin(n,p)$

we can rewrite $P(X \in [a,b])$ as an expectation

$$E[\mathbb{1}_{X[a,b]}] = 1 \cdot \mathbb{P}(X \in [a,b]) + 0(1 - \mathbb{P}(X \in [a,b])$$

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) \quad \text{as } n \to \infty \to a_1$$

According to result from book and CLT that

$$\frac{\sqrt{n}(I_n - I)}{\sigma} \sim N(0,1)$$

so we can write our estimate as,

$$\sqrt{n}\,\frac{\left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) - a_1\right)}{\sigma} \sim N(0,1)$$

essentially we want to estimate 90% CI so

$$0.9 \approx \mathbb{P}\left(\sqrt{n}\,\frac{\left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) - a_1\right)}{\sigma} \in [-1.65, 1.65]\right)$$

$$= \mathbb{P}\left(a_1 \in \left[\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) - \frac{1.65\sigma}{\sqrt{n}}, \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) + \frac{1.65\sigma}{\sqrt{n}}\right]\right.$$

So $a_1 \in \left[\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) - \frac{1.65\sigma}{\sqrt{n}}, \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[a,b]}(X_i) + \frac{1.65\sigma}{\sqrt{n}}\right]$

Part B

3 b

$$a_2 = \frac{1}{n} E\left(\sum_{k=1}^{n} (X_k)^2\right) = E(X_k^2)$$

according to CLT

$$\frac{\sqrt{n}\, E(X_k^2) - a_2}{\sigma} \sim N(0,1)$$

$$0.9 \approx P\left(\frac{\sqrt{n}\, E(X_k^2) - a_2}{\sigma} \in [-1.64, 1.64]\right)$$

$$= P\left(a_2 \in \left[\frac{1}{n}\sum X_k^2 - \frac{1.64\sigma}{\sqrt{n}}, \frac{1}{n}\sum X_k^2 + \frac{1.64\sigma}{\sqrt{n}}\right)\right)$$

so $\mathbb{P}\left(a_2 \in \left[\frac{1}{n}\sum X_k^2 - \frac{1.64\sigma}{\sqrt{n}}, \frac{1}{n}\sum X_k^2 + \frac{1.64\sigma}{\sqrt{n}}\right)\right)$

CI : $a_2 \in \left[\frac{1}{n}\sum X_k^2 - \frac{1.64\sigma}{\sqrt{n}}, \frac{1}{n}\sum X_k^2 + \frac{1.64\sigma}{\sqrt{n}}\right)$

from expo(1) we know $E[X^2] = \frac{2!}{\lambda^n} = 2$

$$\begin{aligned} Var(X^2) &= E[(X^2)^2] - E[X^2]^2 \\ &= E[X^4] - E[X^2]^2 \\ &= \frac{4!}{\lambda^4} - 2 \\ &= 4\cdot 3\cdot 2\cdot 1 - 2 \\ &= 22 \end{aligned}$$

so for close form 22

$$2 \pm 1.65 \cdot \sqrt{22/500}$$

Part C

(a)

Parametric

```
sample_bin<-rbinom(500, 30, 3/4)
estmated_np<-mean(sample_bin)
estmated_p<-estmated_np/30
a<-10
b<-20
sampling_dist<-c()
for (i in 1:1000) {
  sampling_dist<-c(sampling_dist,mean(rbinom(500, 30, estmated_p) %in% c(a:b)))
}
quantile(sampling_dist,.95)
quantile(sampling_dist,.05)
```

```
> quantile(sampling_dist,.95)
  95%
0.222
> quantile(sampling_dist,.05)
   5%
0.164
```

Non-parametric

```
sampling_dist_non<-c()
for (i in 1:1000) {
  bsample <- mean(sample(sample_bin,500,replace=T) %in% c(a:b))
  sampling_dist_non<-c(sampling_dist_non,bsample)

}
quantile(sampling_dist_non,.95)
quantile(sampling_dist_non,.05)
```

```
> quantile(sampling_dist_non,.95)
 95%
0.24
> quantile(sampling_dist_non,.05)
   5%
0.178
```

CLT

```
mean(rbinom(500, 30, 3/4) %in% c(a:b)) +qnorm(0.05)*sd((rbinom(500, 30, 3/4) %in% c(a:b)))/30**0.5
mean(rbinom(500, 30, 3/4) %in% c(a:b)) +qnorm(0.95)*sd((rbinom(500, 30, 3/4) %in% c(a:b)))/30**0.5
> mean(rbinom(500, 30, 3/4) %in% c(a:b)) +qnorm(0.05)*sd((rbinom(500, 30, 3/4) %in% c(a:b)))/30**0.5
[1] 0.07155965
> mean(rbinom(500, 30, 3/4) %in% c(a:b)) +qnorm(0.95)*sd((rbinom(500, 30, 3/4) %in% c(a:b)))/30**0.5
[1] 0.2948693
```

The 90% confidence interval is about [0.072, 0.295]

(b)

Parametric

```
exp_sample<-rexp(500, rate = 1)**2
lamda<-1/mean(exp_sample)
exp_sampling<-c()
for (i in 1:1000) {
  new_sample <- rexp(500, rate = lamda)
  exp_new_sample<-mean(new_sample)
  exp_sampling<-c(exp_sampling,exp_new_sample)

}
quantile(exp_sampling,.95)
quantile(exp_sampling,.05)
```

```
> quantile(exp_sampling,.95)
     95%
1.879856
> quantile(exp_sampling,.05)
      5%
1.620517
```

Non-parametric

```
sampling_expo_non<-c()
for (i in 1:1000) {
  bsample <- mean(sample(exp_sample,500,replace=T))
  sampling_expo_non<-c(sampling_expo_non,bsample)

}
quantile(sampling_expo_non,.95)
quantile(sampling_expo_non,.05)
```

```
> quantile(sampling_expo_non,.95)
     95%
1.969925
> quantile(sampling_expo_non,.05)
      5%
1.507556
```

CLT

```
#clt
#depend on sample
mean(exp_sample) +qnorm(0.05)*sd(exp_sample)/500**0.5
mean(exp_sample) +qnorm(0.95)*sd(exp_sample)/500**0.5
```

```
> mean(exp_sample)+qnorm(0.05)*sd(exp_sample)/500**0.5
[1] 1.508642
> mean(exp_sample)+qnorm(0.95)*sd(exp_sample)/500**0.5
[1] 1.978519
```

The 90% confidence interval is about [1.509, 1.979]

```
#close form
2+qnorm(0.05)*22^0.5/500**0.5
2+qnorm(0.95)*22^0.5/500**0.5
```

```
> 2+qnorm(0.05)*22^0.5/500**0.5
[1] 1.654973
> 2+qnorm(0.95)*22^0.5/500**0.5
[1] 2.345027
```

The 90% confidence interval is about [1.655, 2.345]

Question 4

Part A

4. $\min_{\lambda > 0} \int_0^\infty \log(f_\lambda(x)) f_\lambda(x)\,dx + \int_0^\infty x f_\lambda(x)\,dx$

$= E[\log f_\lambda(x) + x]$

$= E[\log(\lambda) - \lambda x + x]$

$= \log(\lambda) - \lambda E[X] + E[X]$

If $\lambda = 1$

$= 0$ it minie the expesion

Part B

<u>Intuition</u>

b, First set $\theta_0 = 5$

sample $x_1 \dots\dots x_n \sim exp(5)$

Second find $E[Y_\theta] = \frac{1}{n}\sum_{i=1}^{n} (\log(\lambda) - \lambda x + x) \frac{\partial}{\partial\lambda} \log(P_\lambda(x))$

$$= \frac{1}{n}\sum_{i=1}^{n} (\log(\lambda) - \lambda x_i + x_i)\left(\frac{1}{\lambda} - x_i\right)$$

Third pick step size $r_n > 0$

$$\begin{cases} \sum_{n=1}^{\infty} r_n = \infty \\ \sum_{n=1}^{\infty} (r_n)^2 < \infty \end{cases} \quad \text{So } r_n = \frac{c}{n} \quad r_n = \frac{1}{n}$$

Fourth combine everything together.

1, $\theta_0 = 5$

$\theta_{n+1} = \theta_n - r_n \cdot E[Y_\theta]$

<u>Implementation</u>

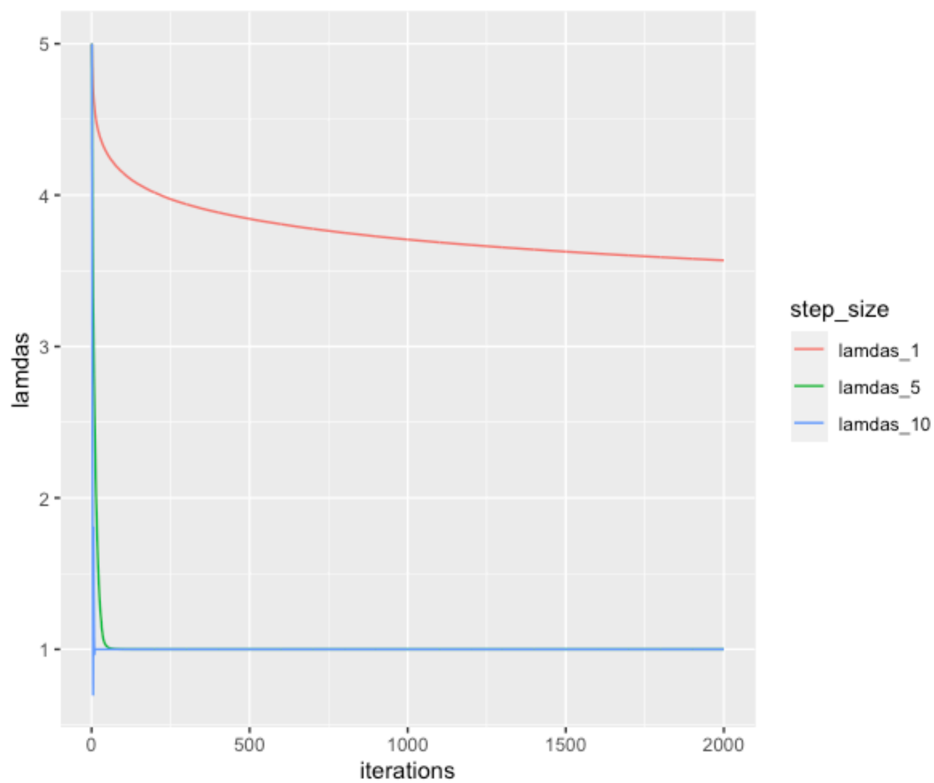
```r
m = 300
lamda=5
question4_x<-rexp(n, rate = lamda)
expected_y_lamda<-function(n,lamda,x){
  sum((log(lamda)-lamda*x+x)*(1/lamda - x))/n
}
lamdas<-c()
step<-1
for (i in 1:2000) {
  question4_x<-rexp(m, rate = lamda)
  lamdas<-c(lamdas,lamda)
  lamda=lamda-step/i*expected_y_lamda(n,lamda,question4_x)

}
```

A side note I repeated this algorithm for different step size and set it accordingly

```
lamdas_1<-lamdas
lamdas_5<-lamdas
lamdas_10<-lamdas
lamdas<-cbind(lamdas_1,lamdas_5,lamdas_10)
lamdas
library(ggplot2)
library("reshape2")
test_data_long <- melt(lamdas, id="lamdas")
colnames(test_data_long)[1]='iterations'
colnames(test_data_long)[2]='step_size'

ggplot(data=test_data_long,
       aes(x=iterations, y=lamdas, colour=step_size)) +
  geom_line()
```



Comment

It seems like the larger the step the faster it converges, however if step is too large then it will drop to quickly and if step is too small it will not/take longer to converge.