Exercies 4 - Alon Goodman & Ran Hassid

Q.1

Trapezoid Rule Function

```
trapezoid_rule <- function(func,up.limit,low.limit,n){
  point <- seq(from = low.limit, to = up.limit, by = ((up.limit - low.limit)/(n-1)))
  s = 0
  for(i in c(1:(n-1))){
    s = s + func(point[i]) + func(point[i+1])
  }
  s = (0.5*(up.limit - low.limit)/(n-1))*s
  print(c("Our Trapezoid estimation is",s),quote = FALSE)
}
```

Simpson Rule Function

```
simpson_rule <- function(func,up.limit,low.limit,n){
  point <- seq(from = low.limit, to = up.limit, by = ((up.limit - low.limit)/(n-1)))
  s = 0
  for(i in c(1:(n-1))){
    s = s + func(point[i]) + 4*func((point[i]+point[i+1])/2) +  func(point[i+1])
  }
  s = ((up.limit - low.limit)/((n-1)*6))*s
  print(c("Our Simpson estimation is",s),quote = FALSE)
}
```

Monte Carlo & CI Function

```
monte_carlo_integration_and_CI <- function(func,up.limit,low.limit,n){
  U <- runif(n = n,min = low.limit ,max = up.limit)
  s = func(U)
  theta = (up.limit - low.limit) * mean(s)
  sd = sqrt((1/n)*var(s))
  u_ci = theta + qnorm(0.975) * sd
  l_ci = theta - qnorm(0.975) * sd
  print(c("Our MCI estimation is",theta),quote = FALSE)
  print(c("Our CI is",l_ci , u_ci),quote = FALSE)
}
```

a

a.1

א.

$$\int_0^1 \frac{\cos(x)}{\sqrt{1-x^2}}\,dx$$

נשים לב ש $x=1$ זו נק' אי הגדרה של הפונקציה.

$$\int_0^1 \frac{\cos(x)}{\sqrt{1-x^2}}\,dx = \begin{bmatrix} t=\sqrt{1-x}\,, & x=1-t^2 \\ dx=-2t\,dt & \end{bmatrix} =$$

$$\int_1^0 \frac{\cos(1-t^2)}{\cancel{\sqrt{1-x}}\,\sqrt{1+x}} \cdot -2\cancel{\sqrt{1-x}}\,dt = \int_1^0 \frac{-2\cos(1-t^2)}{\sqrt{2-t^2}}\,dt = \int_0^1 \frac{2\cos(1-t^2)}{\sqrt{2-t^2}}\,dt$$

וכעת אין נק' בעייתית.

```r
f.a.1 <- function(x)
{
  y <-  (2*cos(1-(x^2)))/(sqrt(2-(x^2)))
  return(y)
}

trapezoid_rule(func = f.a.1,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our Trapezoid estimation is 1.2019697169842
```

```r
simpson_rule(func = f.a.1,up.limit = 1,low.limit = 0,n = 10001)
```

```
## [1] Our Simpson estimation is 1.20196971531721
```

We can see that both method gave us same result.

```r
monte_carlo_integration_and_CI(func = f.a.1,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our MCI estimation is 1.20448909660718
## [1] Our CI is        1.19702763049804 1.21195056271633
```

a.2

$$\int_1^\infty \frac{1}{(1+x^2)^2}\,dx$$

ב.

נשים לב שהבעיה כאן היא אינסוף ולא נוכל להשתמש קירובים של
כמו טרפז וגם סימפסון.

$$\int_1^\infty \frac{1}{(1+x^2)^2}\,dx = \begin{bmatrix} t=\frac{1}{x} \Rightarrow x=\frac{1}{t} \\ dx=-\frac{1}{t^2}\,dt \end{bmatrix} = \int_1^0 \frac{1}{\left(1+\frac{1}{t^2}\right)^2}\cdot -\frac{1}{t^2}\,dt =$$

$$= \int_0^1 \frac{t^4}{(t^2+1)^2}\cdot\frac{1}{t^2}\,dt = \int_0^1 \frac{t^2}{(t^2+1)^2}\,dt$$

ונשאר רק לחשב את:

```
f.a.2 <- function(x)
{
  y <-  (x^2/(1+x^2)^2)
  return(y)
}

trapezoid_rule(func = f.a.2,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our Trapezoid estimation is 0.142699081698723
```

```
simpson_rule(func = f.a.2,up.limit = 1,low.limit = 0,n = 10001)
```

```
## [1] Our Simpson estimation is 0.142699081698724
```

We can see that both method gave us same result.

```
monte_carlo_integration_and_CI(func = f.a.2,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our MCI estimation is 0.142971673004214
## [1] Our CI is          0.141223602280261 0.144719743728168
```

a.3

$$X \sim beta(2,3)$$.ב

$$E\left[\frac{1}{1+\sqrt{x}}\right] = \int_0^1 \frac{1}{1+\sqrt{x}} \cdot f(x)\,dx = \int_0^1 \frac{1}{1+\sqrt{x}} \cdot \frac{1}{B(2,3)} \cdot x^{2-1} \cdot (1-x)^{3-1}\,dx =$$

$$\frac{1}{B(2,3)} \cdot \int_0^1 \frac{x(1-x)^2}{1+\sqrt{x}}\,dx = \frac{\Gamma(2+3)}{\Gamma(2)\cdot\Gamma(3)} \cdot \int_0^1 \frac{x(1-x)^2}{1+\sqrt{x}}\,dx = \frac{4!}{1!\,2!}\int_0^1 \frac{x(1-x)^2}{1+\sqrt{x}}\,dx$$

$$= \int_0^1 \frac{12\,x(1-x)}{1+\sqrt{x}}\,dx$$

```
f.a.3 <- function(x)
{
  y <-  (12*x*(1-x)^2)/(1+sqrt(x))
  return(y)
}

trapezoid_rule(func = f.a.3,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our Trapezoid estimation is 0.628571418600022
```

```
simpson_rule(func = f.a.3,up.limit = 1,low.limit = 0,n = 10001)
```

```
## [1] Our Simpson estimation is 0.628571428568445
```

We can see that both method gave us same result.

```
monte_carlo_integration_and_CI(func = f.a.3,up.limit = 1,low.limit = 0,n = 10000)
```

```
## [1] Our MCI estimation is 0.619963427841418
## [1] Our CI is          0.612173879211468 0.627752976471368
```

a.4

ג.

$X \sim exp(1/4)$, $x \leq 5$

k קבוע שמנרמל את הצפיפות

$$f_X(t) = \begin{cases} k \cdot \frac{1}{4} e^{-1/4\, t} & t \leq 5 \\ 0 & t \geq 5 \end{cases}$$

$$1 = \int_0^5 k\left(\frac{1}{4} e^{-1/4\, x}\right) dx \Rightarrow \frac{4}{k} = \int_0^5 e^{-1/4\, x}\, dx = -4 e^{-1/4\, x} \Big|_0^5 = -4\left(e^{-5/4} - 1\right)$$

$$\Rightarrow \frac{1}{k} = 1 - e^{-5/4} \Rightarrow k = \frac{1}{1 - e^{-5/4}}$$

$$E[X^2 \cdot \sin(X)] = \int_0^5 X^2 \cdot \sin(x) \cdot \frac{1/4\, e^{-1/4\, x}}{1 - e^{-5/4}} dx$$

```
f.a.4 <- function(x)
{
  y <-  ((x^2) * (sin(x)) * (0.25*exp(-0.25*x)) / (1 - exp(-1.25)))
  return(y)
}

trapezoid_rule(func = f.a.4,up.limit = 5,low.limit = 0,n = 10000)
```

```
## [1] Our Trapezoid estimation is -1.58331107373966
```

```
simpson_rule(func = f.a.4,up.limit = 5,low.limit = 0,n = 10001)
```

```
## [1] Our Simpson estimation is -1.58331108105184
```

We can see that both method gave us same result.

```
monte_carlo_integration_and_CI(func = f.a.4,up.limit = 5,low.limit = 0,n = 10000)
```

```
## [1] Our MCI estimation is -1.6044504022532
## [1] Our CI is         -1.62538431141693 -1.58351649308948
```

Q.2

a

א.

$H_0: \lambda = 3 \qquad H_1: \lambda > 3 \qquad \alpha = 0.05 \qquad n = 15 \qquad X \sim Pois(\lambda)$

על פי הקירוב הנורמלי

$$\bar{X} \overset{H_0}{\sim} N\left(\lambda_0, \frac{\lambda_0}{n}\right) \Leftarrow \frac{\bar{X} - \lambda_0}{\sqrt{\frac{\lambda_0}{n}}} \sim N(0,1)$$

לכן נדחה את H_0 אם

$$\frac{\bar{X} - 3}{\sqrt{\frac{3}{15}}} \geq Z_{1-\alpha} = 1.645$$

לכן עבור 15 תצפיות נדחה H_0 אם

$$\bar{X} = 3 + 1.645 \cdot \sqrt{\frac{3}{15}} = 3.73$$

b

```
f.2.b <- function(lambda,n){
  x <- rpois(n = n, lambda = lambda)
  return(((mean(x)-lambda)/sqrt(lambda/n)) > 1.645)
}

result.2.b <- replicate(n = 5000, f.2.b(lambda = 3, n = 15))
alpha_hat.2.monte_carlo <- mean(result.2.b)
sd.2.monte_carlo <- sqrt((1/5000)*var(result.2.b))
CI.2.monte_carlo <- c(alpha_hat.2.monte_carlo-qnorm(0.975)*sd.2.monte_carlo,alpha_ha
t.2.monte_carlo+qnorm(0.975)*sd.2.monte_carlo)
```

```
## [1] Our Monte Carlo estimation for alpha is
## [2] 0.0482
```

```
## [1] Our CI is          0.0422625099889853 0.0541374900110147
```

c

```
f.2.c<- function(lambda,n,known_lambda){
  x <- rpois(n = n, lambda = known_lambda) # importance sampling - pois(lambda from
 a)
  m <- mean(x) >= known_lambda # if lambda = 3 it is relatively rare
  f <- function(x){return(dpois(x, lambda = lambda))} # our target
  g <- function(x){return(dpois(x, lambda = known_lambda))} # our importance sampling
  our_vec <- numeric(n)
  for(i in 1:n){
    our_vec[i] <- f(x[i])/g(x[i])
  }
  return(m*prod(our_vec))
}

result.2.c <- replicate(n = 5000, f.2.c(lambda = 3, n = 15 ,known_lambda = 3.73))
alpha_hat.2.importance_sampling <- mean(result.2.c)
sd.2.importance_sampling <- sqrt((1/5000)*var(result.2.c))
CI.2.importance_sampling <- c(alpha_hat.2.importance_sampling-qnorm(0.975)*sd.2.impor
tance_sampling,alpha_hat.2.importance_sampling+qnorm(0.975)*sd.2.importance_sampling)
```

```
## [1] Our Importance Sampling estimation for alpha is
## [2] 0.0648030928387612
```

```
## [1] Our CI is          0.0623327830914721 0.0672734025860502
```

d

```
sd.2.monte_carlo
```

```
## [1] 0.003029387
```

```
sd.2.importance_sampling
```

```
## [1] 0.001260385
```

We may see that the variance of Monte Carlo estimator is larger than the Importance Sampling estimator. Therefore, we would prefer the Importance Sampling estimator.

Q.3

a

$x_1,\dots,x_n \sim U(0,1)$, $n=1000$ א'

נרצה למצוא את $E(e^{x^2})$ באמצעות אומד MCI שנראה:

כך ש:

$$\theta = E(e^{x^2}) \rightarrow \hat{\theta}_{MCI} = \frac{1}{n}\sum e^{x_i^2}$$

כאשר את x_i נגריל מהתפלגות $U(0,1)$.

```
set.seed(123)
X <- runif(n = 1000,min = 0,max = 1)
g.MCI.x <- exp(X^2)
theta_MCI <- mean(g.MCI.x)
```

```
## [1] Our MCI estimation for E(e^x^2) is 1.45677259715327
```

b

ב)

נתון

$$g_x(x) \propto (x+1)^2 \quad , \quad x \in [0,1]$$

$$\downarrow$$

$$g_x(x) = c \cdot (x+1)^2 \quad , \quad x \in [0,1]$$

נמצא את c:

$$\int_0^1 c \cdot (x+1)^2\, dx = c \cdot \int_0^1 (x^2 + 2x + 1)\, dx = c \cdot \left[\frac{x^3}{3} + \frac{2x^2}{2} + x \Big|_0^1 \right] =$$

$$= c \cdot \left[\frac{1}{3} + 1 + 1 - (0) \right] = 2\frac{1}{3} c \underset{\text{צפיפות}}{=} 1 \longrightarrow c = \frac{3}{7}$$

קיבלנו:

$$g_x(x) = \frac{3}{7}(x+1)^2 \quad , \quad x \in [0,1]$$

$$E_f(e^{x^2}) = \int_0^1 e^{x^2} \cdot 1\, dx = \int_0^1 e^{x^2} \frac{g_x(x)}{g_x(x)}\, dx =$$

$$\int_0^1 \frac{e^{x^2}}{g_x(x)} \underbrace{g_x(x)}_{\substack{\text{צפיפות} \\ 1 = g}}\, dx = E_g\left[\frac{e^{x^2}}{g_x(x)} \right] = E_g\left[\frac{e^{x^2}}{\frac{3}{7}(x+1)^2} \right] = \hat{\theta}_{IS}$$

Screenshot

על מנת לדגום 1000 תצפיות מהתפלגות $g_x(x)$

נשתמש בשיטת ITS:

$$g_x(x) = \frac{3}{7}(x+1)^2$$

$$G_x(x) = \int_0^x \frac{3}{7}(x+1)^2\, dx = \frac{3}{7} \cdot \left[\frac{(x+1)^3}{3} \Big|_0^x \right] = \frac{3}{7}\left[\frac{(x+1)^3 - 1}{3} \right] = \frac{(x+1)^3 - 1}{7}$$

$$\frac{(x+1)^3 - 1}{7} = u$$

$$(x+1)^3 = 7u + 1$$

$$x = (7u + 1)^{\frac{1}{3}} - 1$$

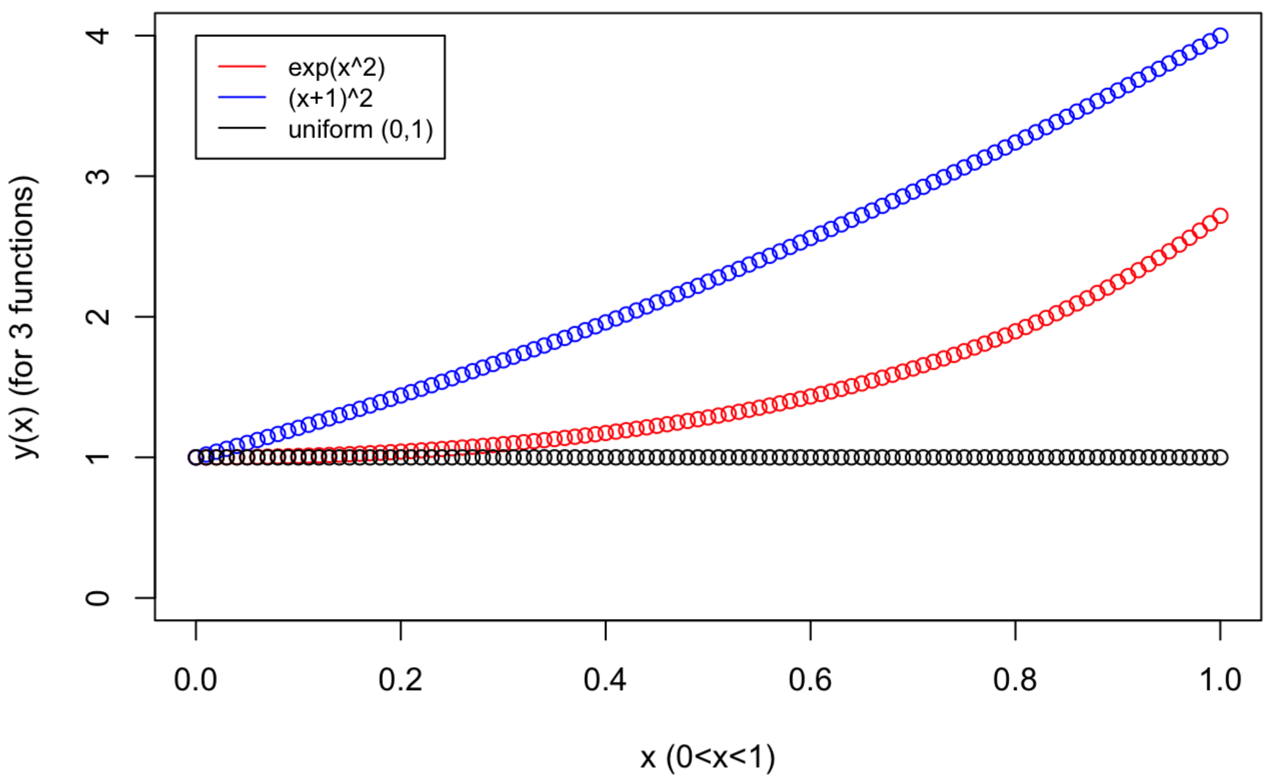
```
set.seed(123)
u.vec <- runif(n = 1000,min = 0,max = 1)
X <- (7*u.vec+1)^(1/3) - 1
g.IS.x <- 7*exp(X^2) / (3*((X+1)^2))
theta_IS <- mean(g.IS.x)
```

```
## [1] Our IS estimation for E(e^x^2) is 1.4616805118491
```

המועמדת היא שגרפה. פונ' דומה לפונ' e^{x^2}.
הגרף ניתן לראות שעבור $x \in [0,1]$ הפונ' $(x+1)^2$
דומה יותר ל- e^{x^2} מאשר התפלגות אחידה $(y=1)$.

```
x.dots <- seq(from = 0, to = 1, by = 0.01)
original_func <- exp(x.dots^2)
g_func <- (x.dots+1)^2

plot(x.dots,original_func,ylim = c(0,4), col = "red", xlab = "x (0<x<1)", ylab = "y
(x) (for 3 functions)")
points(x.dots,g_func, col = "blue")
points(x.dots,rep(1,101), col = "black")
legend(0, 4, legend=c("exp(x^2)", "(x+1)^2","uniform (0,1)"), col=c("red", "blue","bl
ack"), lty = 1 ,cex=0.8)
```



C

```
## [1] The variance theta_MCI is 0.0002216724081349
```

```
## [1] The variance theta_IS is 3.86676245346277e-05
```

We may see that the variance of Monte Caelo Integration estimator is larger than the Importance Sampling estimator. Therefore, we would prefer the Importance Sampling estimator

Q.4

a

```
dbl.exp.sample <- rlaplace(n = 10000,mu = 0,sigma = 1)
```

b

ב׳

נסמן

$$f_x(x) = N(0,1)$$

$$g_x(x) = \text{double exponent}\ (0,1)$$

לפי שיטת קבלה-דחייה נדגום x-ים מהתפלגות g, ונדגום u מהתפלגות אחידה סטנדרטית.

נקבל את הדגימה אם:

$$\frac{f_x(x)}{c\cdot g_x(x)} \geq u$$

אחרת, נדחה את הדגימה.

נמצא את c:

$$c = \max \frac{f_x(x)}{g_x(x)} = \max_x \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}}{\frac{1}{2} e^{-|x|}} =$$

$$= \max_x \underbrace{\sqrt{\frac{2}{\pi}}}_{\text{קבוע}} \cdot e^{|x| - \frac{x^2}{2}} = \max_x \ln e^{|x| - \frac{x^2}{2}} = \max_x |x| - \frac{x^2}{2}$$

נגזור!

$$\frac{\partial\left(|x| - \frac{x^2}{2}\right)}{\partial x} = \begin{cases} 1 - x & x \geq 0 \\ -1 - x & x < 0 \end{cases}$$

נשווה ל-0:

$$x \geq 0: \quad 1 - x \equiv 0 \rightarrow x = 1$$

$$x < 0: \quad -1 - x \equiv 0 \rightarrow x = -1$$

נגזור שוב כדי לבדוק מקסימום:

$$\frac{\partial^2\left(|x| - \frac{x^2}{2}\right)}{\partial^2 x} = \left.\begin{matrix} -1 & x \geq 0 \\ -1 & x < 0 \end{matrix}\right\} \rightarrow$$

שלילי לשני המקרים ולכן $x = 1, -1$ הן מקסימום.

$$c = \frac{f(x^*)}{g(x^*)} = \sqrt{\frac{2}{\pi}} \cdot e^{|x^*| - \frac{x^{*2}}{2}} = \begin{cases} x^*=1 & \sqrt{\frac{2}{\pi}}\, e^{|1| - \frac{1^2}{2}} = 1.315 \\ x^*=-1 & \sqrt{\frac{2}{\pi}}\, e^{|-1| - \frac{(-1)^2}{2}} = 1.315 \end{cases}$$

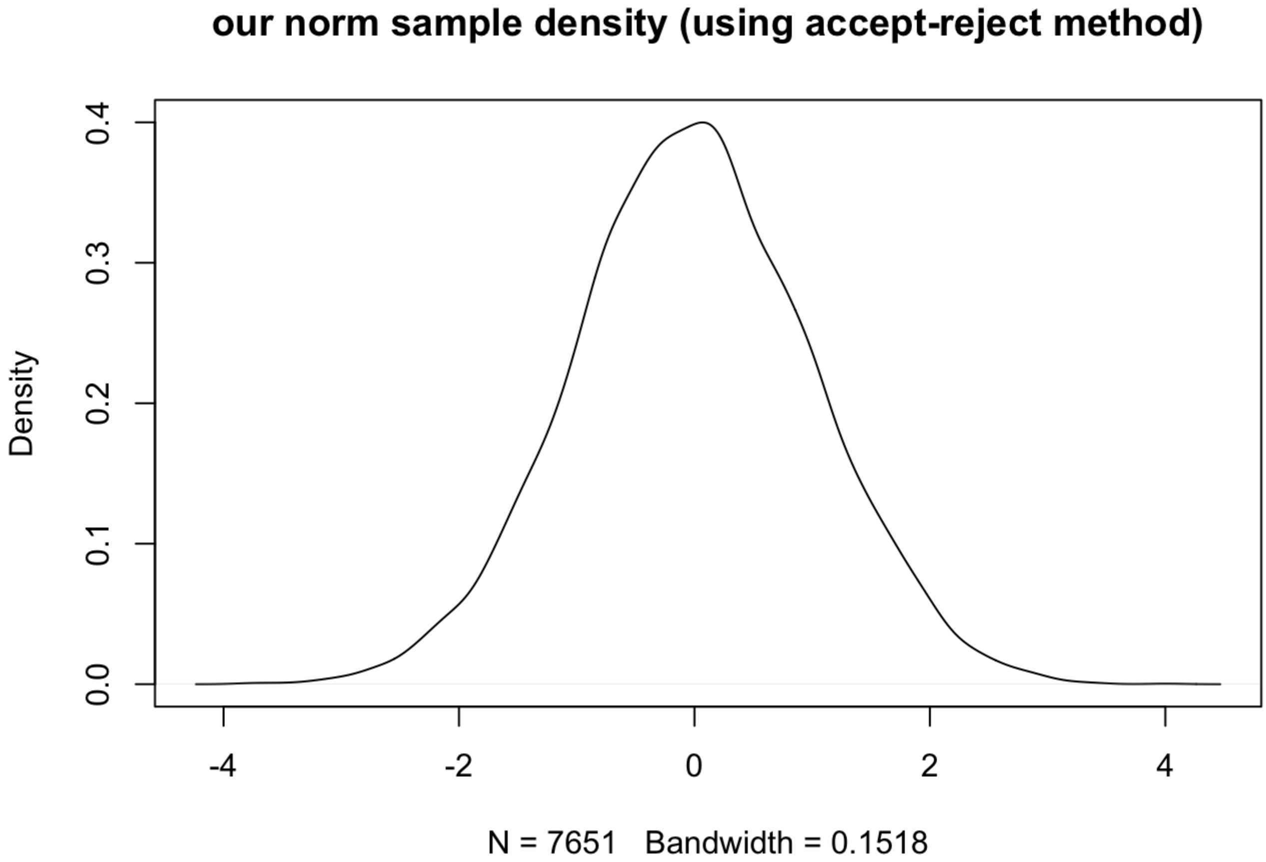
קיבלנו $c \approx 1.315 \ \left(= \sqrt{\frac{2}{\pi}}\, e^{0.5}\right)$

```r
c.Q4 <- sqrt(2/pi) * exp(0.5)
norm.sample <- numeric(0)

for (i in 1:10000) {
  u.Q4.i <- runif(n = 1,min = 0,max = 1)
  f.x.i <- dnorm(x = dbl.exp.sample[i],mean = 0,sd = 1)
  g.x.i <- dlaplace(x = dbl.exp.sample[i],mu = 0,sigma = 1)

  if ((f.x.i/(g.x.i*c.Q4)) >= u.Q4.i) {
    norm.sample <- c(norm.sample,dbl.exp.sample[i])
  }
}
```

```r
plot(density(norm.sample), main = "our norm sample density (using accept-reject metho
d)") #check we got normal distribution
```




```r
Q4.b.data <- cos(norm.sample^3)
```

```
## [1] Estimation for E(cos(x^3)), x~N(0,1) is
## [2] 0.596715077264744
```

C

ג'

$$E_f(\cos x^3) = \int_{-\infty}^{\infty} \cos(x^3) \cdot f_x(x)\, dv =$$

$$= \int_{-\infty}^{\infty} \cos(x^3) \cdot f_x(x) \frac{g_x(x)}{g_x(x)}\, dv =$$

$$= \int_{-\infty}^{\infty} \cos(x^3) \cdot \frac{f_x(x)}{g_x(x)} \cdot g_x(x)\, dv =$$

$$= E_g\left[\cos(x^3) \cdot \frac{f_x(x)}{g_x(x)}\right] = E_g \cos(x_i^3) \sqrt{\frac{2}{\pi}} \cdot e^{|x_i| - \frac{x_i^2}{2}}$$

$$\hat{\theta}_{IS} = \frac{1}{n} \Sigma \cos(x_i^3) \sqrt{\frac{2}{\pi}} \cdot e^{|x_i| - \frac{x_i^2}{2}} \qquad ו \quad x_1, \dots, x_n \sim D.E(0,1)$$

```
Q4.c.data <- cos(dbl.exp.sample^3) * sqrt(2/pi) * exp(-0.5*(dbl.exp.sample^2) + abs(d
bl.exp.sample))
```

```
## [1] Estimation for E(cos(x^3)), x~N(0,1), using IS method is
## [2] 0.601490560059631
```

d

ד'

1) איזה מן האומדים עדיף מבחינת כמות הדגימות מהתפלגות שנתונה?

בשיטת IS אנו דוגמים 10,000 תצפיות מ- D.E
ובעזרן R דוגם מהתפלגות אחידה.
בשיטת קבלה - דחיה אנו משתמשים ב- 10,000 דגימות מ- D.E
ועוד 10,000 דגימות על מנת לבחור בין קבלה לדחיה
של התצפית.

לכן, שיטת IS עדיפה מבחינת דגימה מהתפלגות אחידה.

2) איזה מהאומדים עדיף מבחינת שונות?

```
## [1] The variance theta acc-rej is 5.05271617991582e-05
```

```
## [1] The variance theta_IS is 4.39019083621541e-05
```

We may see that the variance of acc-rej estimator is larger than the Importance Sampling estimator. In this case, sample size is a crucial factor (because in the acc-rej we got much less observations and the division by n/m had great influence). Therefore, we would prefer the Importance Sampling estimator.