1. For $x > 0$ consider the integral

$$J(x) \;=\; \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2}\, dy \;=\; \Phi(x) - 0.5 \;=\; \frac{1}{2} P(|Z| \le x) \quad \text{where } Z \sim N(0,1)$$

(a) Use a single sample of size 1,000 from a $N(0,1)$ distribution to estimate $J(x)$ at $x = (0.10,\ 0.33,\ 0.67,\ 1.00, 1.40,\ 1.70,\ 1.97,\ 2.2,\ 2.5)$. For comparison, compute also the values using the R-function **pnorm()**. **Hint:** Use the **ecdf()** function in R applied to the absolute values of your sample.

(b) For a given x we could also estimate $J(x)$ using samples from the uniform distribution on $[0, x]$, $U[0, x]$, but it would be a nuisance to have to do this uniform sampling for every x. By making the transformation $t = y/x$ show that

$$J(x) \;=\; \frac{1}{\sqrt{2\pi}} \int_0^1 x \,\exp(-(xt)^2/2)\, dt$$

(c) Use this result to estimate $J(x)$ at the x-values of part (a) using a single sample of size 1000 from a $U(0, 1)$ distribution. Compare with the results of part (a).

(a)

```
```{r}
x <- c(0.10, 0.33, 0.67, 1.00, 1.40, 1.70, 1.97, 2.2, 2.5)
0.5*(pnorm(x,0,1)-pnorm(-x,0,1))

z <- abs(rnorm(1000,0,1))
cdf <- ecdf(z)
0.5*(cdf(x)-cdf(-x))

```
```

```
[1] 0.03982784 0.12930002 0.24857110 0.34134475 0.41924334 0.45543454 0.47558081 0.48609655 0.49379033
[1] 0.0385 0.1375 0.2540 0.3435 0.4190 0.4530 0.4700 0.4845 0.4970
```

We can see it's a pretty good estimation that two group of values are close to each other.

(b) $J(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{y^2}{2}} dy$

We know $t = \frac{y}{x}$, so $y = tx$.

$J(x) = \frac{1}{\sqrt{2\pi}} \int_0^{\frac{x}{x}} e^{-\frac{(tx)^2}{2}} dt \cdot x$

$J(x) = \frac{1}{\sqrt{2\pi}} \int_0^1 x \cdot e^{-\frac{(xt)^2}{2}} dt$

(c)

```
```{r}
result <- rep(NA,9)
for(i in 1:9) {
  xi <- x[i]
  J=function(t) {
    return(1/sqrt(2*pi)*xi*exp(-(xi*t)^2/2))
  }
  t <- runif(1000,0,1)
  eva_uni <- J(t)
  result[i] <- mean(eva_uni)
}
result

```
```

```
[1] 0.03982803 0.12931011 0.24848654 0.34259718 0.41871122 0.44908416 0.47873435 0.48697341 0.50677553
```

It's also a good estimation, which is very close to the estimation from part a.

2. Compute the integral

$$\int_0^{\infty} \frac{1}{x^{10} + 3x + |\cos(x)|} \mathrm{d}x$$

using (i) deterministic integration using the R function **integrate()**
and (ii) Monte Carlo integration. In your Monte Carlo, use realizations from an exponential distribution with rate parameter equal to 0.5. Use 10,000 Monte Carlo realizations.
Report the Monte Carlo error and 95% confidence interval for your Monte Carlo estimate.

```{r}
f_x = function(x){
  return(1/(x^10+3*x+abs(cos(x))))
}
integrate(f_x, 0, Inf)

n = 10000
expRate = 0.5
expSample = rexp(n, rate=expRate)
(mcarloInt = mean(f_x(expSample)/dexp(expSample, rate=expRate)))
(mcarloError = sd(f_x(expSample)/dexp(expSample, rate=expRate))/sqrt(n))
(c(mcarloInt - 1.96*mcarloError, mcarloInt + 1.96*mcarloError))

```

```
0.5220095 with absolute error < 9.7e-05
[1] 0.516208
[1] 0.005928091
[1] 0.5045890 0.5278271
```

3. Use Monte Carlo to compute the integral

$$\int_{-\infty}^{\infty} \frac{1}{1+\exp(x^2)}\, dx$$

using realizations from a standard normal $N(0,1)$ distribution. Use 1,000 Monte Carlo realizations to estimate the mean and the Monte Carlo error.

````
```{r}
N = 1000
f_x = function(x){
  return(1/(1+exp(x^2)))
}
x_norm_1 = rnorm(N,0,1)
eva_norm_1 = f_x(x_norm_1)/dnorm(x_norm_1,0,1)
mean(eva_norm_1)
sd(eva_norm_1)/sqrt(N)
```
````

```
[1] 1.065703
[1] 0.009374413
```

4. Repeat the previous question, instead using a Normal distribution

   (a) with standard deviation 0.8, $N(0, 0.8^2)$

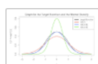
   (b) with standard deviation 0.5, $N(0, 0.5^2)$

   In each case give the mean and Monte Carlo error estimates.

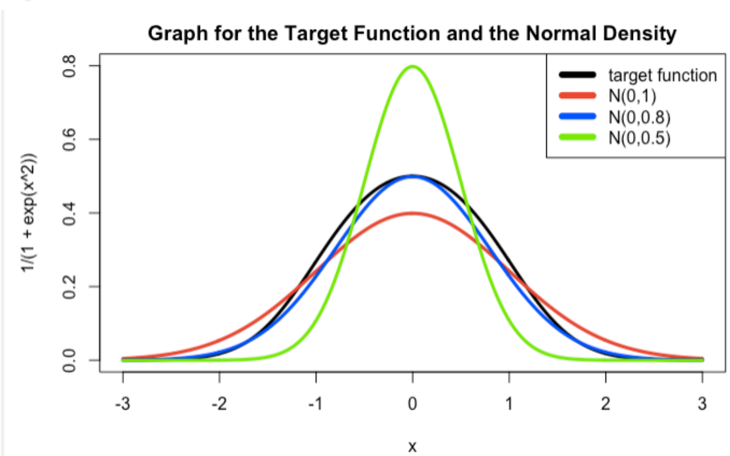
   Plot the four curves, the target function $1/(1 + \exp(x^2))$ and the three Normal densities, over the range -3 to 3. Explain why the estimate using SD=0.8 gives the most precise estimate of the integral.

```{r}

f_x_2 = function(x){
  return(1/(1+exp(x^2)))
}
x_norm_2 = rnorm(N,0,0.8)
eva_norm_2 = f_x_2(x_norm_2)/dnorm(x_norm_2,0,0.8)
mean(eva_norm_2)
sd(eva_norm_2)/sqrt(N)


f_x_3 = function(x){
  return(1/(1+exp(x^2)))
}
x_norm_3 = rnorm(N,0,0.5)
eva_norm_3 = f_x_3(x_norm_3)/dnorm(x_norm_3,0,0.5)
mean(eva_norm_3)
sd(eva_norm_3)/sqrt(N)


x=seq(-3,3,0.01)
plot(x,y=1/(1+exp(x^2)),xlim=c(-3,3), ylim=c(0, 0.8),type="l",lwd=3,main="Graph for the Target Function and
the Normal Density")
lines(x=seq(-3,3,0.01),dnorm(x,mean=0,sd=1),type="l",col="red",lwd=3)
lines(x=seq(-3,3,0.01),dnorm(x,mean=0,sd=0.8),type="l",col="blue",lwd=3)
lines(x=seq(-3,3,0.01),dnorm(x,mean=0,sd=0.5),type="l",col="green",lwd=3)
legend("topright",c("target function","N(0,1)","N(0,0.8)","N(0,0.5)"),col=c("black","red","blue","green"),lw
d=6)
```

R Console



```
[1] 1.06974
[1] 0.002816315
[1] 1.024623
[1] 0.03410886
```



Using SD=0.8 is the best, because we can see from the graph that the blue curve is the closest distribution to the black distribution.