# 1. Question 1

## 1.1. a.

### 1.1.1. *Output.*

```
> em(estimate_pi=FALSE)
[1] "Iteration 1.000000: lambda 3.264076"
[1] "Iteration 2.000000: lambda 4.556793"
[1] "Iteration 3.000000: lambda 4.865708"
[1] "Iteration 4.000000: lambda 4.900126"
[1] "Iteration 5.000000: lambda 4.903378"
[1] "Iteration 6.000000: lambda 4.903680"
[1] "Iteration 7.000000: lambda 4.903708"
[1] "Iteration 8.000000: lambda 4.903711"
[1] "Iteration 9.000000: lambda 4.903711"
[1] "Iteration 10.000000: lambda 4.903711"
$lambda
[1] 4.903711

$pi
[1] 0.5
```

## 1.2. b.

### 1.2.1. *Output.*

```
> em(estimate_pi=TRUE)
[1] "Iteration 1.000000: lambda 3.264076"
[1] "Iteration 1.000000: pi 0.579319"
[1] "Iteration 2.000000: lambda 4.416869"
[1] "Iteration 2.000000: pi 0.499359"
[1] "Iteration 3.000000: lambda 4.846927"
[1] "Iteration 3.000000: pi 0.479270"
[1] "Iteration 4.000000: lambda 4.906107"
[1] "Iteration 4.000000: pi 0.476781"
[1] "Iteration 5.000000: lambda 4.912155"
[1] "Iteration 5.000000: pi 0.476530"
[1] "Iteration 6.000000: lambda 4.912750"
[1] "Iteration 6.000000: pi 0.476506"
[1] "Iteration 7.000000: lambda 4.912808"
[1] "Iteration 7.000000: pi 0.476503"
[1] "Iteration 8.000000: lambda 4.912814"
[1] "Iteration 8.000000: pi 0.476503"
[1] "Iteration 9.000000: lambda 4.912814"
[1] "Iteration 9.000000: pi 0.476503"
[1] "Iteration 10.000000: lambda 4.912814"
[1] "Iteration 10.000000: pi 0.476503"
$lambda
[1] 4.912814

$pi
```

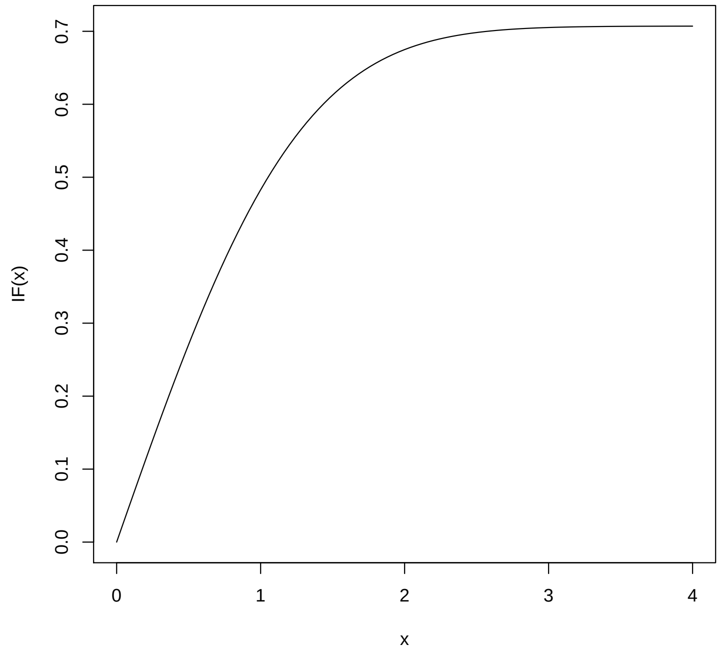
```
[1] 0.4765031
```

### 1.3. Code.

```
# Statistical inference Assignment 2 Question 1 ----
# Simulate the data, assuming Pi is known ----
set.seed(2012)
pi = .5
true_lambda = 5
n = 1000
x = rep(NA, n)
for (i in 1:n) {
  x[i] = rexp(1, true_lambda)
  if (rbinom(1, 1, .5) == 1) {
    x[i] = x[i] + 1
  }
}
hist(x, breaks=100)

# EM algorithm ----
em = function(estimate_pi=FALSE)
{
  # Initialise ----
  iterations = 10
  lambda = rep(NA, iterations)
  lambda[1] = 1/mean(x)
  for (k in 1:iterations) {
    # E step ----
    e = rep(NA, n)
    for (i in 1:n) {
      num = pi * dexp(x[i], lambda[k])
      if ((x[i] - 1) <= 0) {
        denom = pi * dexp(x[i], lambda[k])
      } else {
        denom = pi * dexp(x[i], lambda[k]) + (1 - pi) * dexp(x[i] - 1, lambda[k])
      }
      e[i] = num/denom
      #print(sprintf("i %f x[i] %f num %f denom %f e[i] %f", i, x[i], num, denom, e[i]))
    }
    # M step ----
    #sum(e*x)
    #sum((1-e)*(x-1))
    lambda[k + 1] = n/(sum(e*x) + sum((1-e)*(x-1))) # Sum part is wrong?
    print(sprintf("Iteration %f: lambda %f", k, lambda[k + 1]))
    if (estimate_pi==TRUE) {
      pi = sum(e)/n
      print(sprintf("Iteration %f: pi %f", k, pi))
    }
  }
  return(list(lambda=lambda[k], pi=pi))
}
em(estimate_pi=FALSE)
em(estimate_pi=TRUE)
```

Figure 1. Influence function for $\hat{\theta}$



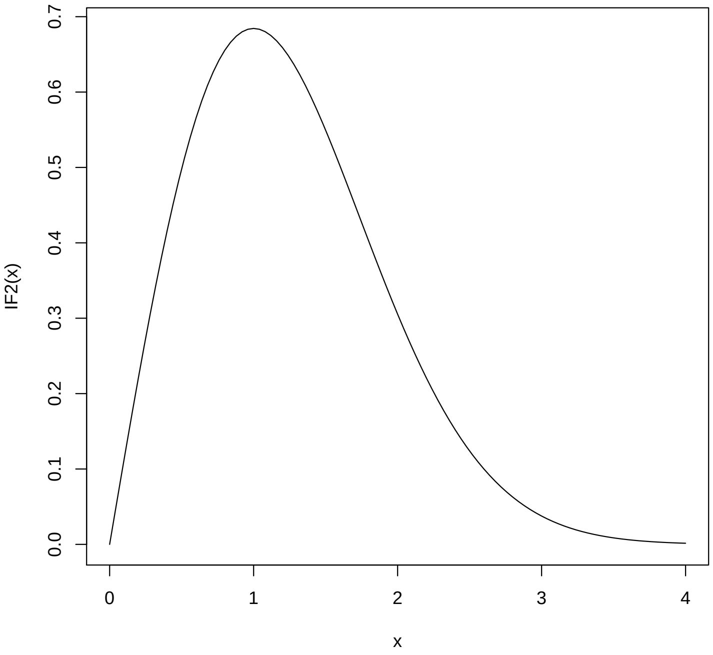
## 2. Question 2

### 2.1. **Code.**

```
# Question 2 ----
# Visualising influence functions
# (e) ----
IF = function(y)
{
  multiplicand = 2*sqrt(pi)
  return(multiplicand*(pnorm(y) - .5))
}
pdf("influence_function1.pdf")
curve(IF, xlim=c(0, 4))
dev.off()

IF2 = function(y)
{
  multiplicand = 4*sqrt(pi)
  return(multiplicand*y*dnorm(y))
}
pdf("influence_function2.pdf")
curve(IF2, xlim=c(0, 4))
```

Figure 2. Influence function for $\tilde{\theta}$



```
dev.off()
```