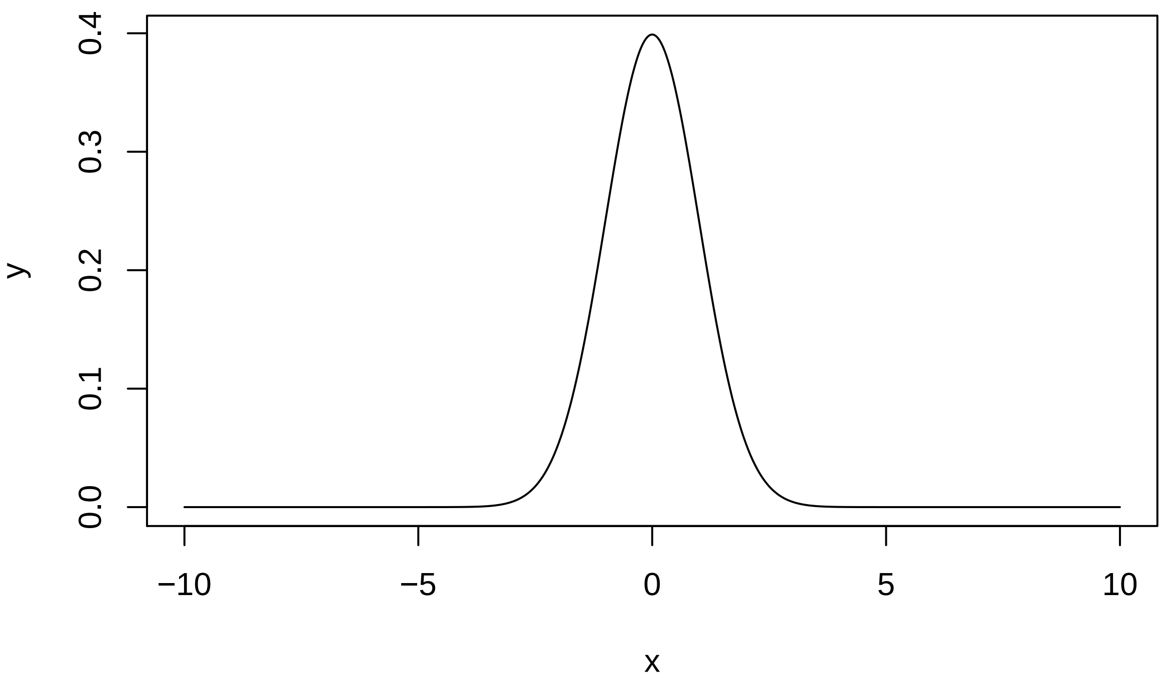
Homework 1

Andre Guimaraes Duarte

July 18, 2016

Question 2

```
x <- seq(-10, 10, length = 1000) # range of x can be modified
y <- dnorm(x, 0, 1, FALSE) # generate normal random variable values for each x
plot(x, y, type = "l") # plot to verify that p(x)>=0 for every x
```




```
# Function that calculates a rough integral under a curve,
# given x and y values. It breaks the area under the curve into
# many rectangles, and sums the total area of these rectangles.
# The more x values in the range (length), the better the approximation
integ <- function(x, y){
  area <- 0
  for (i in seq(length(x)-1)) {
    # width of the rectangle is one step of x
    # height of the rectangle is the mean of current and next y
    area <- area + ((x[i+1]-x[i])*(y[i+1]+y[i])/2)
  }
  return(area)
}

# Print the value of the integral of the pdf of the normal random distribution (should be 1)
print(integ(x, y))
```

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

```r
# verification
print(integrate(dnorm, -10, 10))
```

```
## 1 with absolute error < 7.4e-05
```

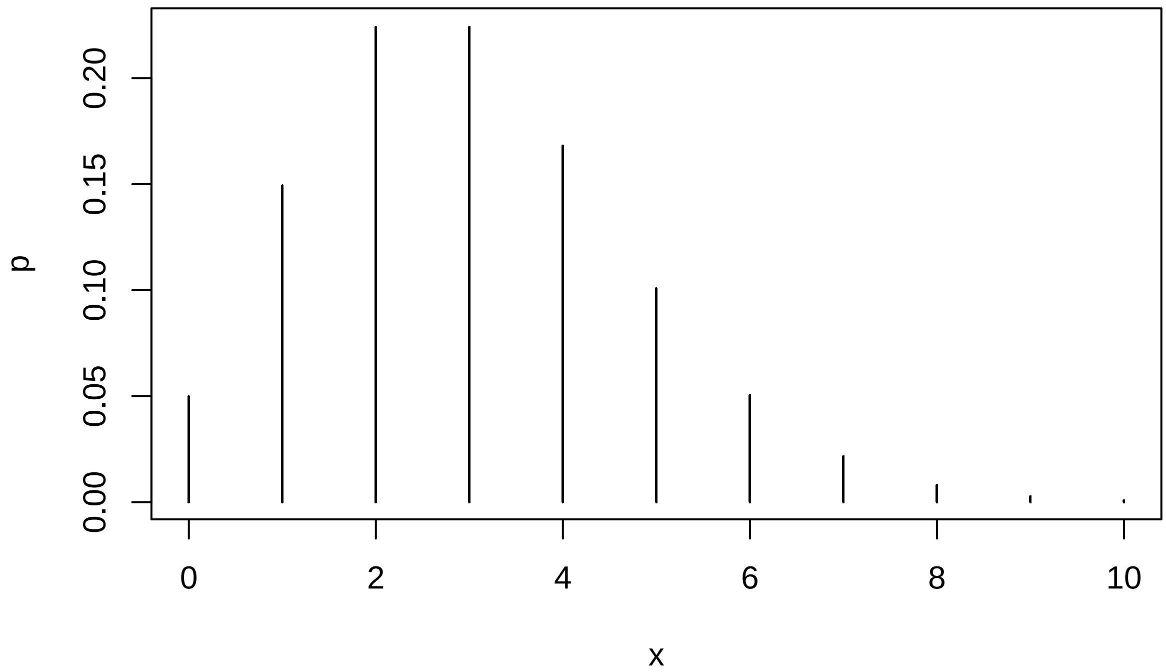
Question 16

```r
#(lambda^x * exp(-lambda))/(x!)
#p(x+1) = (lambda/(x+1))*p(x)
lambda <- 3
n <- 1000
p <- vector()
x <- vector()

# Function that generates a random variable that is governed by
# the Poisson distribution. It exploits the recursive relationship
# p(x+1) = (lambda/x+1)*p(x)
pois <- function(x){
  # Case p(0)
  if(x==0){
    return(exp(-lambda))
  }
  # Recursion!
  return((lambda/x)*pois(x-1))
}

# 1000 realizations
for(i in 1:n){
  # Track every x and its associated p(x)
  x[i] <- sample(0:10, 1) # the range for the values of x can be modified
  p[i] <- pois(x[i])
}

# Plot the probabilities against the values x takes
plot(x, p, type = "h")
```


0.00
0.05
0.10
0.15
0.20
p
0
2
4
6
8
10
x