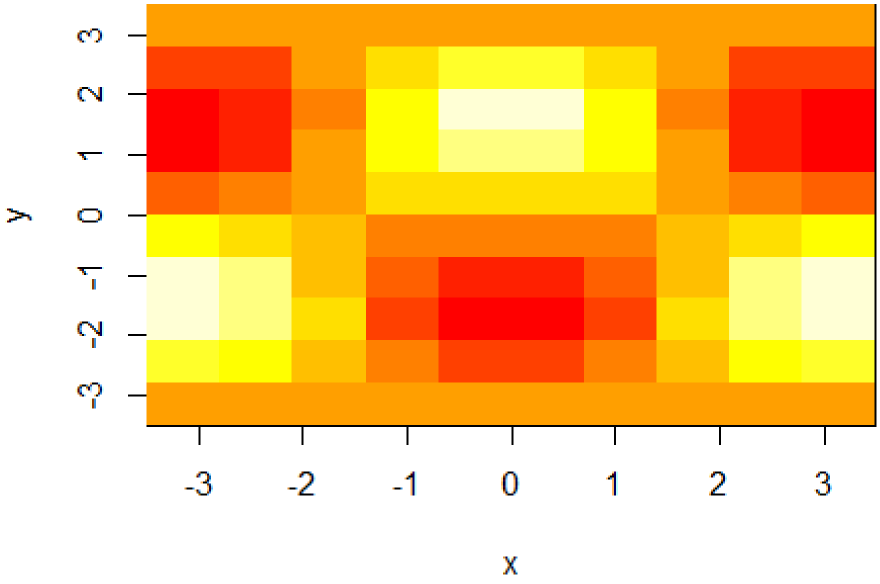
Assignment 1

FNU Anirudh

September 9, 2015

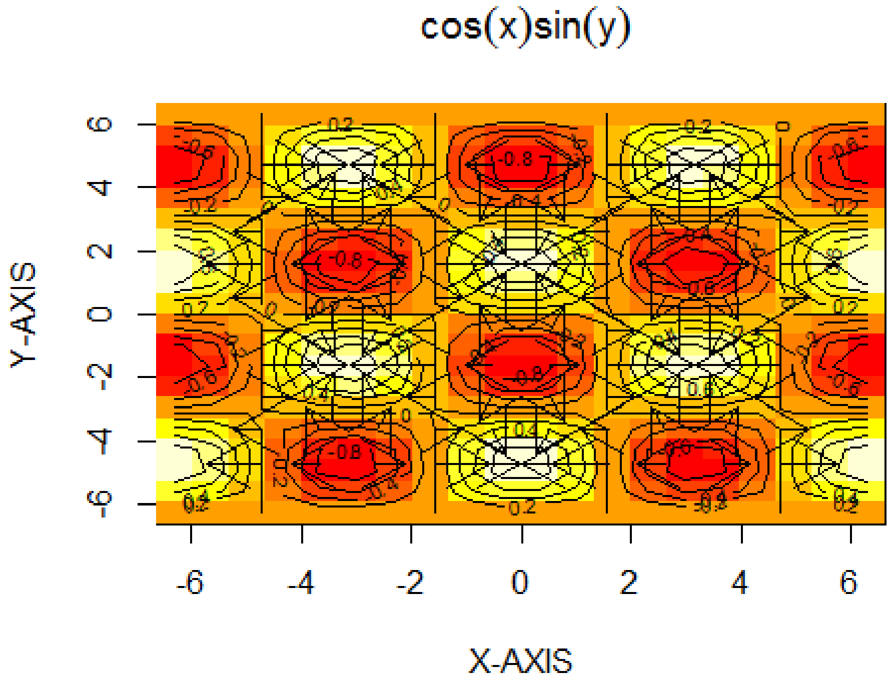
Question 1

```
plot_image=function(x,y)
{
  value_range=expand.grid(x,y)
  z_matrix=matrix(cos(value_range$Var1)*sin(value_range$Var2),nrow=length(x))
  image(x,y,z_matrix)
}
x=seq(-pi,pi,length = 10)
y=x
plot_image(x,y)
```



Question 2

```
plot_image = function(ex,x,y,arrowsx, arrowsy)
{
  # the Derivative function
  f = deriv(ex, c("x", "y"), function(x, y){})
  # Grid values of X and Y
  grid = expand.grid(x,y)
  # get the values of function for given X and Y
  z = f(grid[,1], grid[,2])
  # Matrix Form
  m = matrix(z, nrow = length(x), ncol = length(y))
  #Image Plot
  image(x,y, m, xlab = "X-AXIS", ylab = "Y-AXIS",
        main = ex)
  #Contour plot on the image plot itself
  contour(x,y,m, add = TRUE)
  # grid values to plot the gradient
  arrowsgrid = expand.grid(arrowsx, arrowsy)
  # call the func passing the arrow grid value
  grad = f(arrowsgrid[,1],arrowsgrid[,2])
  # get the gradient values
  grad = attr(grad,"gradient")
  grad_tip = arrowsgrid + grad
  # get the X and Y gradient
  # Add Arrows to the plot
  arrows(arrowsgrid[,1], arrowsgrid[,2], grad_tip[, 1],
         grad_tip[,2])
}
# define range for X
x = seq(-2*pi, 2*pi, length.out = 20)
# make Y equal to X
y = x
# define the X range values for plotting arrows
arrows_x = seq(-1.5*pi,1.5*pi, length.out = 10)
# make Y range equal to X range
arrows_y = arrows_x
# Expression
bivarient_expression = ~cos(x)*sin(y)
# Call Function
plot_image(bivarient_expression, x, y, arrows_x, arrows_y)
```



Question 3

```
a=c(0.12,0.15,0.15,0.10,0.13,0.15,0.14,
    0.08,0.11,0.09,0.14,0.09,0.13,0.14,
    0.12,0.16,0.15,0.13,0.12,0.12,0.09)
b=c(88,66,71,63,101,55,76,
    49,63,38,91,79,41,36,
    73,55,42,49,50,90,51)
stem(a,scale=2)
```

```
##
##   The decimal point is 2 digit(s) to the left of the |
##
##    8 | 0
##    9 | 000
##   10 | 0
##   11 | 0
##   12 | 0000
##   13 | 000
##   14 | 000
##   15 | 0000
##   16 | 0
```

```
stem(a,scale=3)
```

```
##
##   The decimal point is 2 digit(s) to the left of the |
##
```

```
##     8 | 0
##     9 | 000
##    10 | 0
##    11 | 0
##    12 | 0000
##    13 | 000
##    14 | 000
##    15 | 0000
##    16 | 0
```

```
stem(a,scale=4)
```

```
## 
##   The decimal point is 2 digit(s) to the left of the |
## 
##     8 | 0
##     8 | 
##     9 | 000
##     9 | 
##    10 | 0
##    10 | 
##    11 | 0
##    11 | 
##    12 | 0000
##    12 | 
##    13 | 000
##    13 | 
##    14 | 000
##    14 | 
##    15 | 0000
##    15 | 
##    16 | 0
```

(i):-Inference (Scale=2)

1.Not Symmetrical 2.No Outlier 3.Bell Shaped

(ii):- Inference (Scale=3)

1. Not Symmetrical 2.No Outlier 3.L Shaped

(iii):- Inference (Scale=4)

1. Symmetrical
2. No outlier
3. Bell shaped

(b) leaf unit: 1

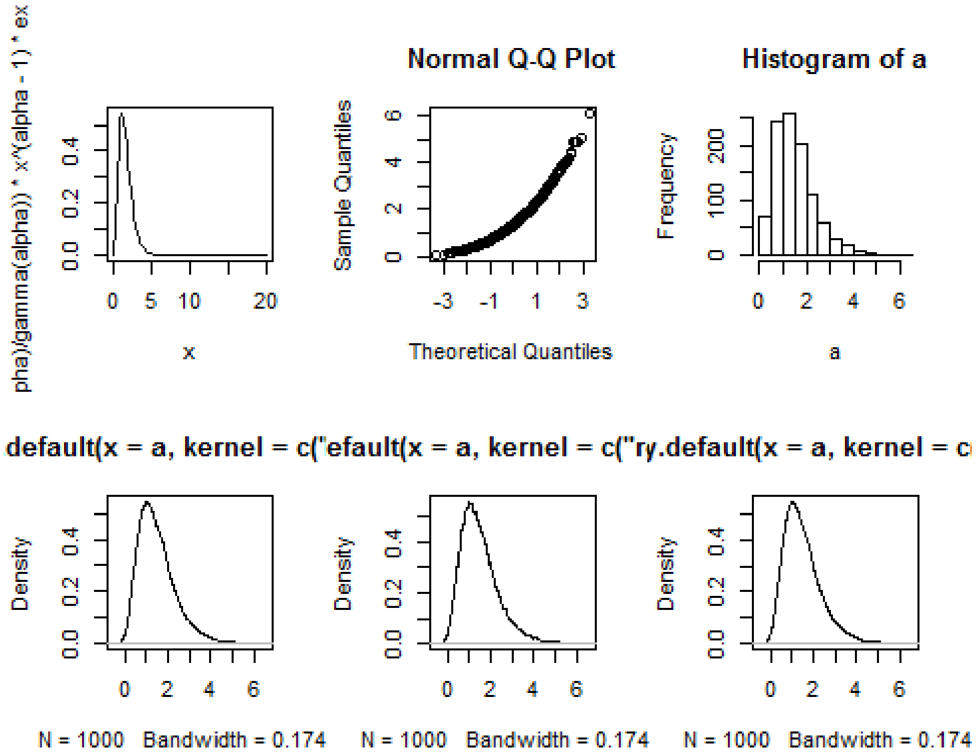
```
     n:  21

    3.|  68
    4*| 12

    4.| 99

    5*| 01

    5.| 55

    6*| 33

    6.| 6

    7*| 13

    7.| 69

    8*|

    8.| 8

    9*| 01

    9.|
```

10*| Infernce:-

1. L- Shaped.
2. 101 is outlier.
3. There are more values on top hence it is not symmetrical.

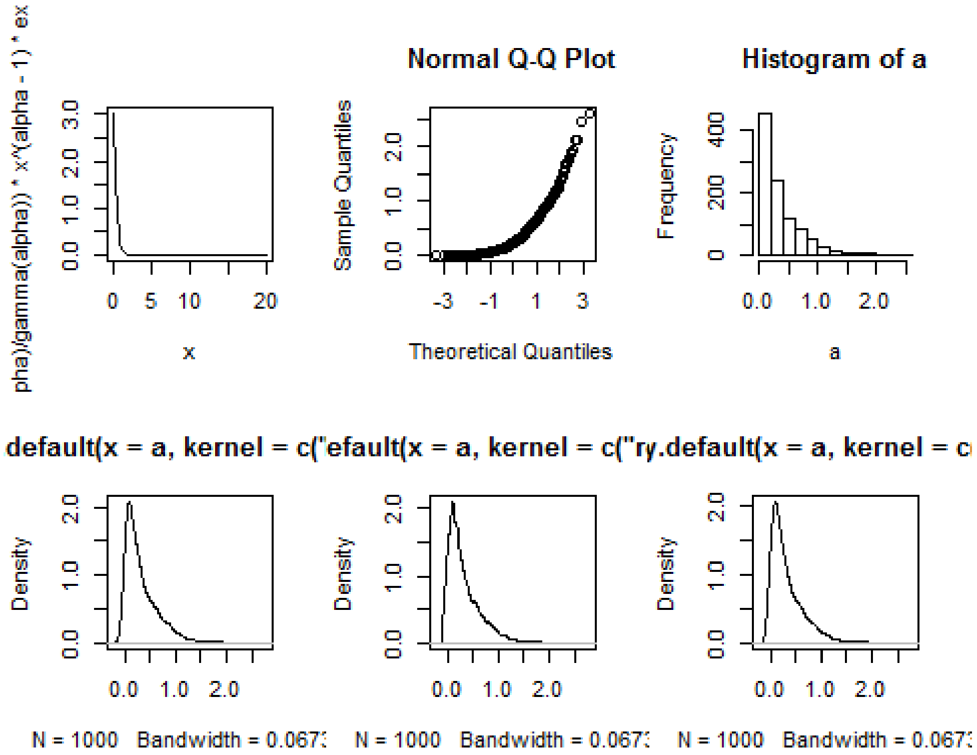
Question 4

```r
alpha = 3
beta = 2
# Density function
par(mfrow = c(2, 3))
curve((((beta^alpha)/gamma(alpha))*x^(alpha - 1)*exp(-beta*x)), 0, 20)
# Data generated from RGamma
a = rgamma(1000, shape = alpha,rate = beta)
# QQ Plot for a
qqnorm(a)
# Histogram
hist(a)
plot(density(a, kernel = c("gaussian")))
plot(density(a, kernel = c("rectangular")))
plot(density(a, kernel = c("cosine")))
```



Second Value

```
alpha = 1
beta = 3
# Density function
par(mfrow = c(2, 3))
curve((((beta^alpha)/gamma(alpha))*x^(alpha - 1)*exp(-beta*x)), 0, 20)
# Data generated from RGamma
a = rgamma(1000, shape = alpha,rate = beta)
# QQ Plot for a
qqnorm(a)
# Histogram
hist(a)
plot(density(a, kernel = c("gaussian")))
plot(density(a, kernel = c("rectangular")))
plot(density(a, kernel = c("cosine")))
```



Gaussian function bests represents the data.