Statistics 6260/4960

In Class Work: October 25

The function prototype is:

in.circle <- function(pts,cntr,r)

pts is an nX2 matrix. Each row of the matrix represents a point in a two-dimensional space. The first column contains the x coordinates of the points and the second column contains the y coordinates of the points. There are n points.

cntr is a vector of length 2. It is the center of a circle. cntr[1] is the x coordinate of the center of the circle, and cntr[2] is the y coordinate of the center of the circle.

r is the radius of the circle. It has length 1.

Find which points in the pts matrix are in the circle, and which are outside the circle. A point exactly on the circle is considered in the circle.

Return a matrix containing the points in the circle. If there are no points in the circle return NULL.

The function will also produce a plot. The circle should be plotted with type = 'l'. Look at the plotting script to see how to plot a circle. The points in the circle should be plotted using a red color. The points outside the circle should be plotted using a blue color.

Be sure to set the xlim and ylim values so that all points in the pts matrix and the entire circle will be displayed in the plot. This will involve finding the min and max coordinates in the pts matrix. Also you must find the min and max coordinates of the circle (these can easily be calculated). Once you have this information, set xlim and ylim accordingly.

Be sure to have error checks on your input arguments.

Here is an example run:

```
> set.seed(40)

> x = runif(14,0,10)

> x = round(x,3)

> y = runif(14,0,10)

> y = round(y,3)

> pts = cbind(x,y)

> print(pts)
```

```
       x     y
 [1,] 6.836 9.595
 [2,] 8.729 7.951
 [3,] 6.901 3.866
 [4,] 1.159 9.763
 [5,] 1.950 0.953
 [6,] 4.612 5.819
 [7,] 2.035 4.723
 [8,] 5.908 3.023
 [9,] 3.739 1.103
[10,] 1.413 5.718
[11,] 0.962 7.907
[12,] 7.029 6.383
[13,] 0.776 3.113
[14,] 2.349 6.927
>
> cntr = c(4,5)
> r    = 3.5
>
> a = in.circle(pts,cntr,r)
> print(a)
       x     y
[1,] 6.901 3.866
[2,] 4.612 5.819
[3,] 2.035 4.723
```

[4,] 5.908 3.023

[5,] 1.413 5.718

[6,] 7.029 6.383

[7,] 2.349 6.927

The plot is shown here