User Guide

Juliana Caamaño, George Perez, Jacqueline Goldstein
10-Aril-2012

I.Introduction

Finding the right emotions can be quite a strenuous endeavor and as students, who has time? So why not let your favorite computer program – R – do it for you? This user guide will give you step by step instructions on how to make R be a source of emotions other than anger and frustration.

1. How to create the Happy Face

- We need to create a vector to specify where in the Cartesian plane we would like to plot the eyes; in this case we will use the coordinates (3.5,30) and (6.5,30).

  > eyesx<-c(3.5,6.5) → points in the x-axis
  > eyesy<-c(30,30) → points in the y-axis

- After creating the vectors we are ready to plot the points. We will use the command *plot(x,y,…)* which is the generic function for plotting R objects.

  > plot(eyesx,eyesy,xlim=c(-10,10),ylim=c(-40,40),pch=19,cex=1.5,col="blue")

  The elements *xlim* and *ylim* correspond to the desired range in the plotting window; since we want to be able to see all four quadrants of our graph, we have created a vector from -10 to 10 for the x-axis and from -40 to 40 for the y-axis. The element *cex* corresponds to a numerical value that expands (or magnifies) the plotted points relative to its default size; in this case we choose *cex=1.5*. And the last command *col* corresponds to the color we would like to give to our plotted points; with this command you can write the desired color inside quotation marks or as a number (see color chart in R help).

- We are ready now to create the smile. Notice that the "happy" smile is parabola, for this reason we will create a function to be plotted in the plane.

  >happy<-function(x) (2*x-10)2+10**

  *Here, **happy** is given the value of a function of x, hence we write *function(x)* and following the mathematical function for the parabola.*

- In order to plot our function in the plane, we will use the command *curve(…)*.

  >curve(happy,from=3.3,to=6.7,col="red",add=T)

  *Since we would like the smile to be plotted in a specific space on the Cartesian plane, we use *from* and *to* which corresponds to the domain of the function (x values.*

- To fill our happy smile we will use the command *polygon(…).*

  First, we need to specify the points to be filled (points in the x-axis)

  >x<-seq(3.3,6.7,0.01)

  → the points are a sequence from 3.3 to 6.7 (same domain of our function) and goes in increments of 0.01.

  > happyfilled<-(2*x-10)2+10**

  → corresponds to the area to be filled in our plot (notice is the same original function)

  > polygon(x,happyfilled,col=2)

  → first element corresponds to the values on the x-axis and the second elements is corresponds to the values on the y-axis. Notice that for this command we used *col=2* which corresponds to the color red.

- To be able to write text on our plot we will use the command *text(…).*

  >text(5,40,label="Happy")

2. How to create the Confused Face

- To keep the graph asymmetric, we are going to keep the eyes on the same plane as the others.

  ```
  > confusedx <- c(-6.5,-3.5)
  > confusedy<-c(30,30)
  ```

- Next, we plot:

  ```
  > points(confusedx,confusedy, pch=”?”,cex=1.5,col=”blue”)
  ```

- And the text to label the face

  ```
  >text(-5,40,label=”Confused”)
  ```

  Both functions are lower level functions so “add=T” is not necessary

- For the mouth we can use either a sine or cosine function shifted up, to the left, and with shorter periods.

  ```
  >confused<- function (x) cos(6*x)+15
  > curve(confused, from=-7.2, to=-2.7, add=T)
  ```

3. How to Create the Greedy Face

- To create the eyes we will follow the same procedure as the previous emotions. The coordinates in this case will be (-6.5,-3.5) and (-10,10).

  ```
  > dollarx<-c(-6.5,-3.5)
  > dollary<-c(-10,-10)
  ```

- To plot these points in the plane we will use the command *points(x,y,…).*

  ```
  >points(dollarx,dollary,pch="$",cex=1.5,col="green")
  ```

  → first and second element correspond to the desired coordinates.

- There are several ways to create circles in R, in this case we will use the command *points.* Points allows us to add in a shape, such as a circle with this precise elements.

  ```
  > points(-5,-25,pch=19,cex=4,col=3)
  ```

  →The first and second elements correspond to the coordinates of the center of the circle. The idea is that we use the original pch notation or a circle but enlarge it and change its color to create a large green circle.

- Our GREEDY emotion has been created, now we give it its name.

```
> text(-5,-40,label="Greedy")
```

4. How to Create the Sad Face

- In order to begin, start out by setting the coordinates for the eyes. At this point the Cartesian Plane should already have the scale set from the previous command in the Happy face section above.

  Input the values:

  ```
  >sadx<-c(3.5,6.5)
  >sady<-c(-10,-10)
  ```

- These commands will allow you to set the points for the Sad face's eyes. After this we would like to input command *points*, which allows us to actually graph the points on the quadrant. Pay attention to the command written for it includes small changes to color and shape from the previous commands you have already inputted .

  ```
  >points(sadx,sady,pch=19,cex=1.5,col="grey")
  ```

- After this command is inputted correctly you should see two grey eyes appearing on your fourth quadrant plane. Now, we work on making a sad mouth.
- This is easily created my setting a function that will shape as a parabola facing downwards, and setting it precisely under the eyes.

  For this function we input:

  ```
  >sad<-function(x) -((x-5)**2)-20
  ```

- Then from here, we will graph this line using *curve*, since *plot* has already been used to set up the plane, a lower level command is necessary to keep adding to the same graph.

```
>curve(sad,from=3.3,to=6.7,col="blue",lwd=5,add=T)
```

- After inputting this, the parabola should appear under the eyes. Pay attention to the *add=T* command, this is important in order to add the curve to the current graph.
  Now, it is time to input the text, "SAD" to label the last of your quadrant planes.

```
>text(5,-40,label="Sad")
```

- To add lines to separate the quadrants, use abline.

```
> abline(h=0,v=0)
```

- This will give you a label under your sad face, completing the sad face for your game.

5. How to set-up the Game:

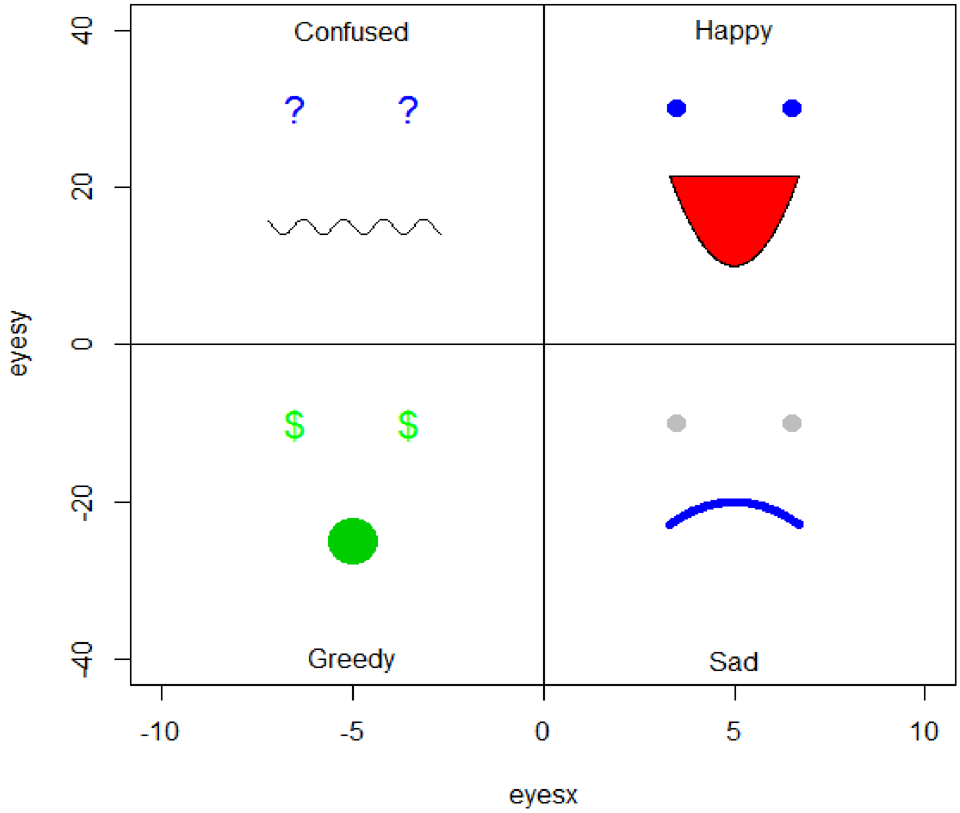
- The game portion utilizes the a random number generator in R – *runif* – that we will use to randomly pick a point within the four quadrants.

```
>Choose<-matrix(runif(2,-10,-10),1,2)
>points(choose,pch=19)
```

- "runif(2,-10,10)" tells R to pick two values between -10 and 10. "matrix(…,1,2)" tells R to put the two runif values into a one by two matrix. These serve as the x and y values.

II. Common Mistakes

1. Forgetting the add=T to high level functions that need to be added to existing graphs.
2. Forgetting the letter "c" in the command sadx=*c*(3.5,6.5), used in forming vectors.
3. Using sufficient parenthesis in functions to help R read the function properly, leading to a graph you exactly want.


Confused
Happy
? ?
$ $
Greedy
Sad
40
20
0
-20
-40
eyesy
-10
-5
0
5
10
eyesx