

Graphics 101 in R

Song Liu (song.liu@bristol.ac.uk)

GA 18, Fry Building,

Microsoft Teams (search "song liu").

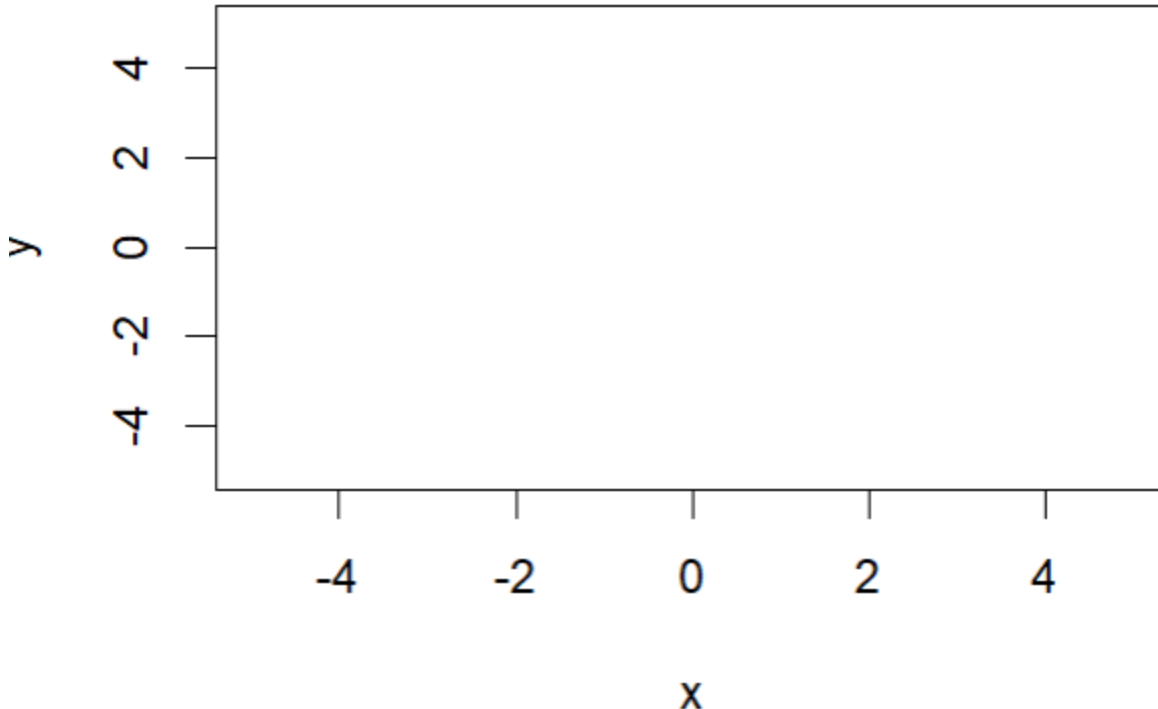
Graphics in R

- R has a powerful graphics features, which allows users to visualize data easily.
 - Hint: You can type `demo("graphics")` in the command line to see demo code/plots.
- You may have already seen some of the graphics function, so I will be brief.

Create a plot: `plot`

`plot` function is usually used to create an empty canvas, ready for further plotting actions.

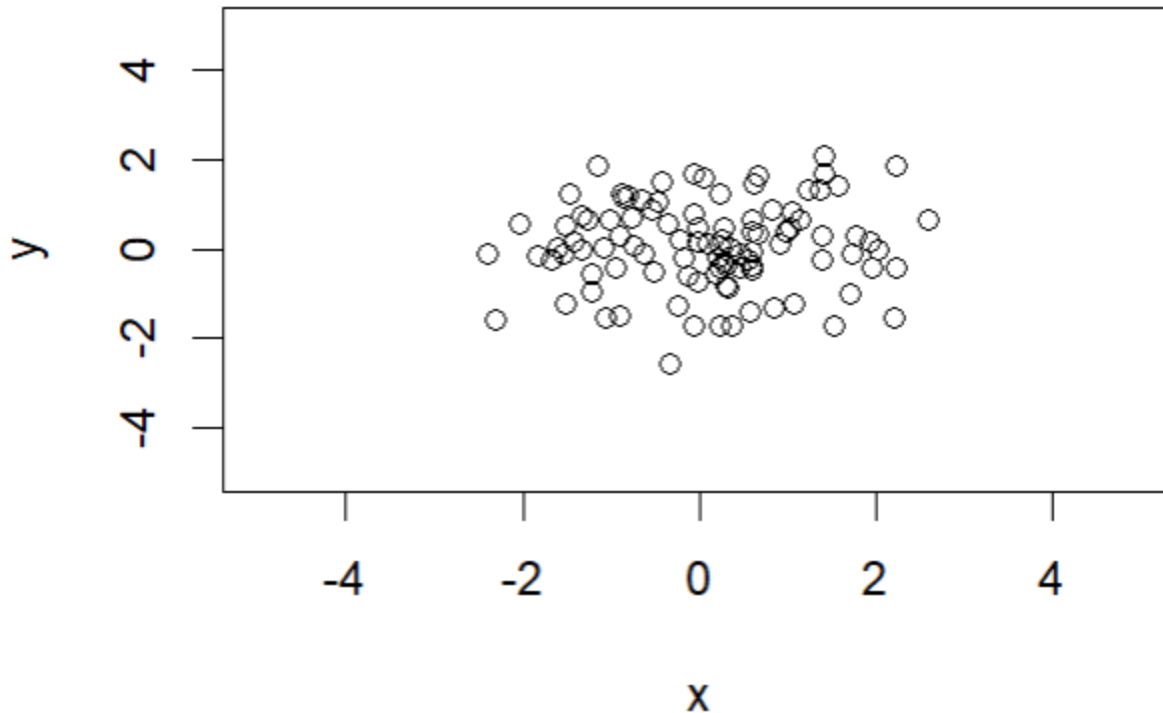
```
# create an empty plot, with x axis ranges from -5 to 5  
# y axis ranges from -5 to 5.  
plot(c(-5,5),c(-5,5), type = "n", xlab = "x", ylab = "y")
```



Visualize Data Points

`points` can be used to draw data points on a 2D plot.

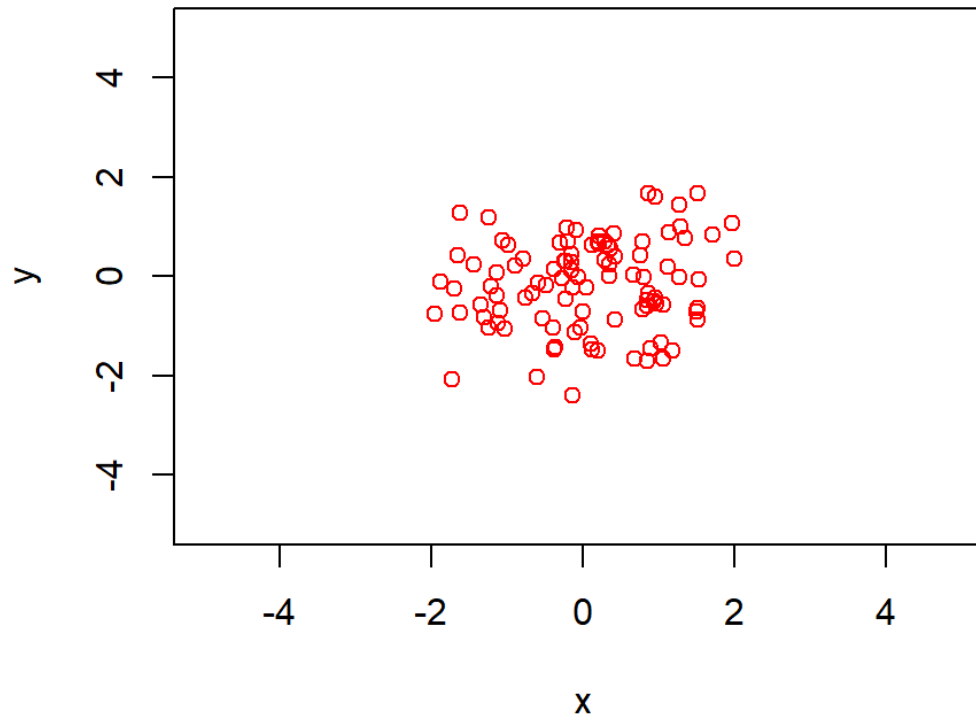
```
x <- rnorm(100)
y <- rnorm(100)
# draw points on the current plot. The first dimension
# of points are stored in a vector x and the second
# dimension are stored in a vector y.
points(x,y)
```



Visualize Data Points

You can change the the color of plotted points

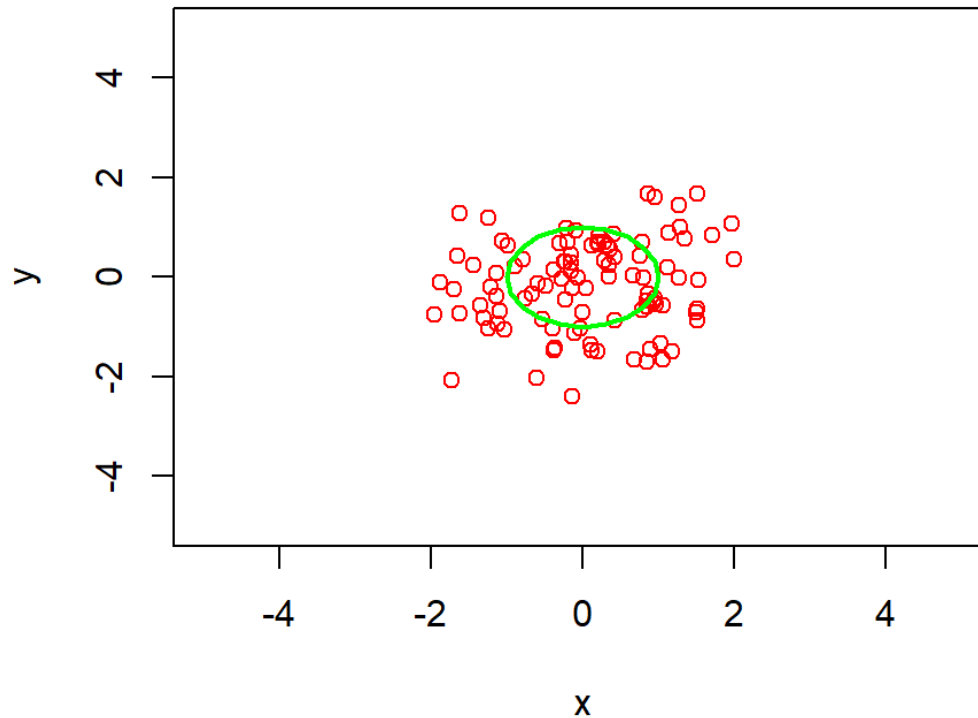
```
points(x,y,col="red")
```



Draw Lines

You can draw line in a 2D plot using `lines` function:

```
# generate points on a circle  
circ <- cbind(cos(0:20*pi/10), sin(0:20*pi/10))  
# draw lines by connecting these points  
lines(circ[,1],circ[,2], col = "green", lwd = 2)
```



Homework

This homework is inspired by a scene in the TV show "The Office":

<https://www.youtube.com/watch?v=QOtuX0jL85Y>

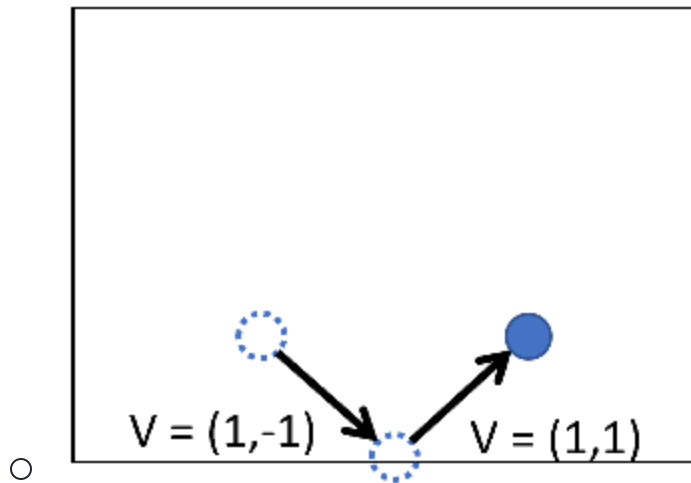
We are going to write an animated visualization that mimics the DVD logo bouncing around the edge of the TV.

Homework

0. Read lecture slides on visualizations carefully.
1. Create an empty plot, whose x-axis and y-axis range from -5 to 5.
2. Initialize two vectors:
 - $x = (0, 0)$ and $v = (.23456, .12345)$.
 - Draw the vector x as a **red** dot in the plot.
 - Add plot title: "point simulation, press ESC to stop."
 - Hint: `?title`

Homework

3. Now let us write a function that updates the point x 's location based on the velocity vector v .
- If the position of x is slightly out of the box, the point will be **bounced back**.



Homework

3. Write a function `update(x,v)` that takes two vectors `x` and `v` as inputs and returns the updated `x` and `v`.
- First, the function checks if `x` is outside of the boundary.
 - If so, it updates the velocity vector by "reflecting it". See the picture in the previous slide.
 - Then, update `x <- x + v`
 - Finally, `return(list(x,v))`
 - The `list` combines two vectors and allows you to return multiple values from a function.
 - Hint: `x[2] > 5 | x[2] < -5` is true if `x` is above the ceiling or is below the floor.

Homework (submit)

4. Your code should look like this:

```
update <- function(x,v){  
  # TODO: Check the boundary collision, and update v.  
  
  x <- x + v #update x  
  return(list(x,v))  
}  
  
x <- c(0,0) #the initial position of x  
v <- c(.23456,.12345) #the initial velocity of x  
  
while(T){  
  #TODO: create a new plot and draw x's current location  
  
  xv <- update(x,v) # update x and v  
  # getting new x and v from the returned list.  
  # list indexing uses double brackets [[]]  
  x<- xv[[1]]  
  v<- xv[[2]]  
  
  # wait a bit to allow RStudio plot the picture.  
  Sys.sleep(.1)  
}
```

Homework (submit)

- Your plot should look like this.
https://github.com/anewgithubname/MATH10017-2022/blob/main/lecs/lab14_1.mp4
- In the TV show, each time the logo bounces, it changes color randomly. Could you revise your code so that the point in your plot also changes its color every time it bounces?