Matemáticas Computacionales
Práctica 1: Gráficas de curvas en R

Profesor: Ángel Isabel Moreno Saucedo
Semestre Febrero - Junio 2021

1. Introducción

En esta primera práctica se hará una de las cosas básicas al momento de aprende R. Se repasaran las curvas en $\mathbb{R}^2$ vistas en primer semestre en la materia de Geometría Analítica[1]. Se graficarán curvas como la recta, parábola, circunferencia, elipse e hipérbola.

2. Curvas de $\mathbb{R}^2$

2.1. Línea recta

Partiendo de la ecuación general de la linea recta

$$Ax + By + C = 0, \tag{1}$$

se graficará en R. La ecuación (1) se utilizara en su forma pendiente intersección para poder codificar más facil.

$$y = mx + b \tag{2}$$

Hay que pensar en lo datos que ocupamos para gráficar una recta que son la pendiente m y la intersección en el ejes de las ordenadas b. Estos dos datos son suficientes para gráficar la recta. Ahora veamos algo de código.

```r
m <- 3 #pendiente
b <- 2 #interseccion

#funcion de la linea recta
f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-5, 5, 1)#vector de -5 a 5
y <- f(m, b, x) #evaluamos

plot(x, y) #graficamos
```

Cuadro 1: Primer código en R para gráficar una recta.

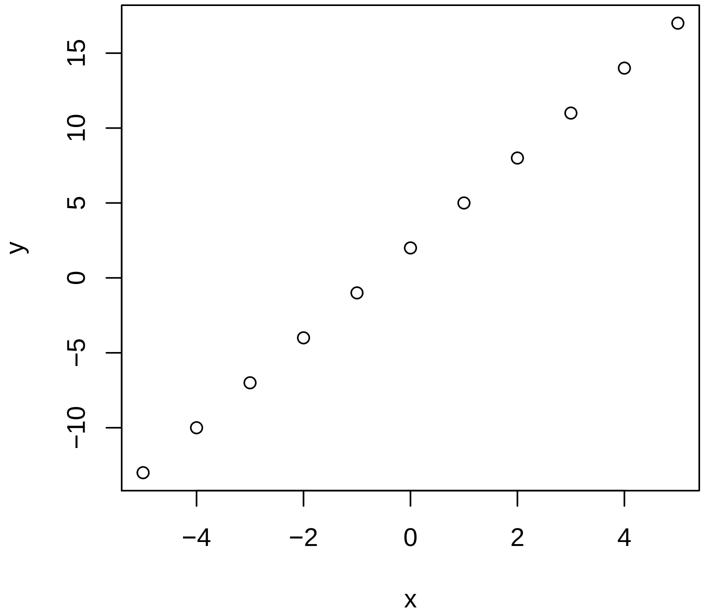

Figura 1: Gráfico que arroja el primer código ejecutado

Si ejecutamos el código anterior en R obtenemos la figura (1). Observe que solo tenemos graficados puntos, esto es por que la función `plot()` tiene por default gráficar puntos. Para que nos gráfique líneas tenemos que ponerle la opción “`l`”que es de `line`, ya que andamos ahí vamos a agregar los nombres a los eje y las líneas para generar el plano cartesiano y con más puntos en las x.

```
#Linea recta
m <- -2 #pendiente
b <- -2 #interseccion

#funcion de la linea recta
f <- function(m, b, x){
  return(m * x + b)
}

x <- seq(-5, 5, 0.01)#vector de -5 a 5
y <- f(m, b, x) #evaluamos

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que pasa por el 0 en las y
```

Cuadro 2: Código actualizado en R para gráficar una recta.

Agregando como argumento `type = ‘‘l’’, xlab = ‘‘Eje X’’, ylab = ‘‘Eje Y’’` en la función `plot()` dibuja la gráfica con lineas en lugar de puntos, Eje X como nombre en el eje de las x y Eje Y como nombre del eje Y. Esto se muestra en la figura (2)

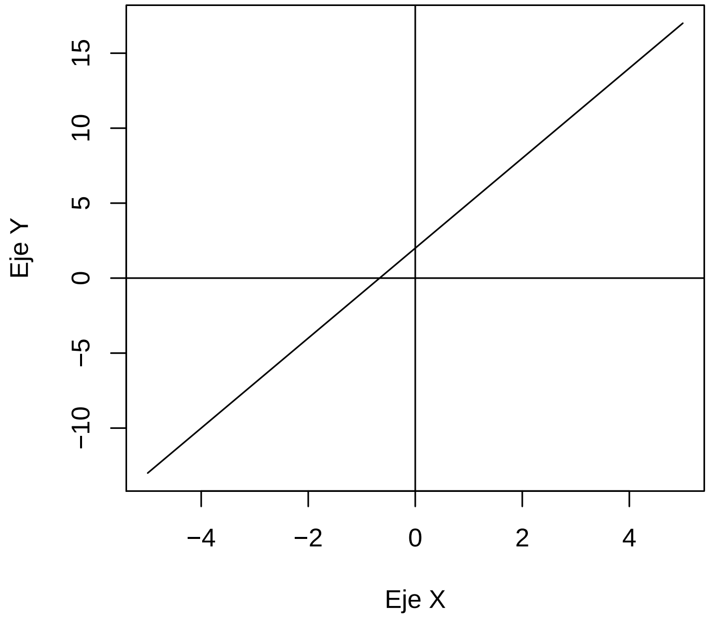

Figura 2: Línea recta con pendiente 3 y pasa por la intersección en 2.

2.2. Parábola

Para gráficar la parábola lo haremos de otra forma distinta que sera utilizando la ecuación (3), si forma general. Veamos como lo codificaremos.

$$y = Ax^2 + Bc + C \tag{3}$$

```
#Parabola
g <- function(x){
  return(2*x^2 + x - 2)
}

x <- seq(-5, 5, 0.01)#vector de -5 a 5
y <- g(x)

plot(x, y, type = "l", xlab = "Eje X", ylab = "Eje Y") #graficamos
abline(h = 0, v = 0) #una linea horizontal que pasa por el 0 en las x y una linea vertical que
    pasa por el 0 en las y
```

Cuadro 3: Código en R para gráficar una parábola.

El código no cambia mucho al de la recta, se utiliza las misma funciones. La figura (3) muestra la gráfica de la ecuación $y = 2x^2 + x - 2$.

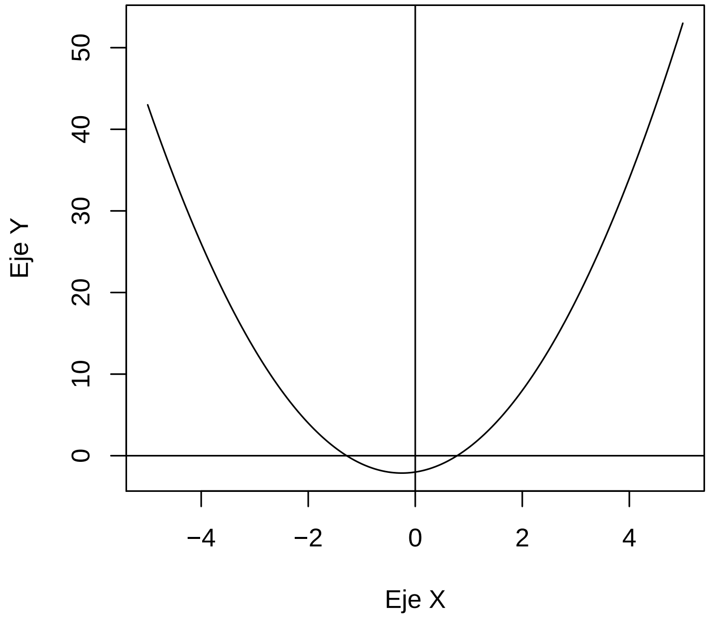

Figura 3: Grafica de la ecuacion $y = 2x^2 + x - 2$.

2.3. Circunferencia

Dada la ecuación ordinaria de circunferencia

$$(x - h)^2 + (y - k)^2 = r^2, \tag{4}$$

donde (h, k) es el centro y r es el radio. Utilizando estos datos graficaremos la circunferencia. Primero la ecuación (4) la despejaremos con respecto a y obteniendo las dos ecuaciones:

$$y = k \pm \sqrt{r^2 - (x - h)^2}, \tag{5}$$

restringiendo el dominio en $x \in [h - r, h + r]$. Se codifica una función que reciba todos estos datos como entrada y arroje como salida la gráfica de la circunferencia con centro en (h, k) y radio r.

El código del cuadro (4) se programa una función llamada `circunferencia(h, k, r)`. Note que se agrega una validación para $r > 0$ y sí $r = 0$ entonces la gráfica es un punto en (h, k), en otro caso es una circunferencia con el dominio: $x \in [h - r, h + r]$. Se evalúa en en dominio dado obteniendo `ypositiva` y `ynegativa` con las ecuaciones en (5). Se agrega adicionalmente dos argumentos mas `xlim = c(h - (r + 1), h + (r + 1))` y `ylim = c(k - (r + 1), k + (r + 1))` en la función `plot()` para controlar los limites de los ejes coordenados, la función `lines(x, ynegativa, type = ''l'')` añade en el gráfico existente la linea generada por los vectores $(x, ynegativa)$ y `points(x = h, y = k, col = ''red'')` agrega el centro de la circunferencia con color rojo. La figura (4) muestra la salida del código.

```r
#Circunferencia
circunferencia <- function(h, k, r){
  if (r >= 0){ # r tiene que ser positivo
    if (r == 0){ # si es r = 0, entonces es un punto
      plot(x = h, y = k, xlab = "Eje X", ylab = "Eje Y") # grafica del punto
    } else{
      x <- seq(h - r, h + r, 0.01) # ya que no podemos graficar en todo R^2
      ypositiva <- k + sqrt(r^2 - ((x - h)^2)) # parte positiva de la circunferencia
      ynegativa <- k - sqrt(r^2 - ((x - h)^2)) # parte negativa de la circunferencia
      # graficamos primero la parte positiva
      plot(x, ypositiva, type = "l", xlim = c(h - (r + 1), h + (r + 1)), ylim = c(k - (r + 1),
           k + (r + 1)),
           xlab = "Eje X", ylab = "Eje Y")
      lines(x, ynegativa, type = "l") # agregamos la parte negativa
      abline(h = 0, v = 0) # agregamos los ejes
      points(x = h, y = k, col = "red") # dibujamos el centro
    }
  } else{
    return(print("El radio no es positivo."))
  }
}

# ejecutamos la funcion
circunferencia(2, -3, 3)
```

Cuadro 4: Código actualizado en R para gráficar una circunferencia.

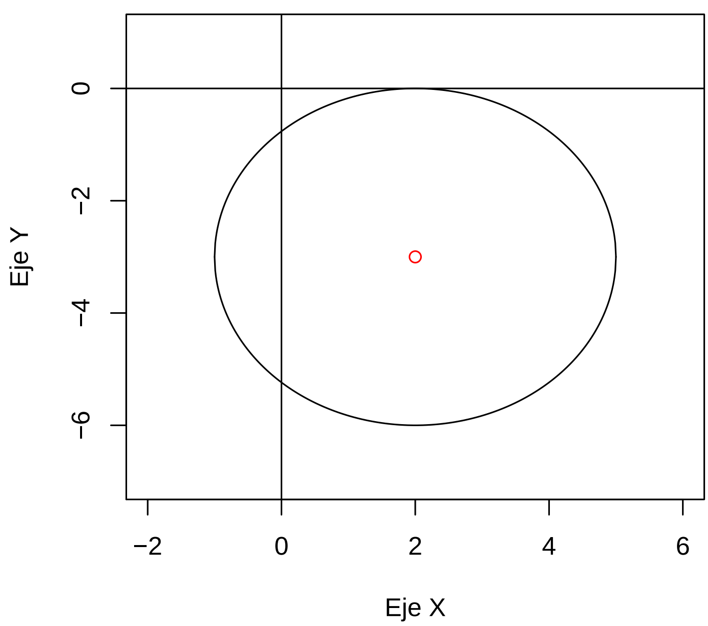

Figura 4: Grafica de una circunferencia con centro en (2, -3) y radio 3.

2.4. Elipse

Partiendo de la ecuación ordinario de la elipse se obtiene despejando con respecto a y:

$$y = k \pm \sqrt{b^2 - \frac{b^2}{a^2}(x-h)^2} \tag{6}$$

con dominio $x \in [h-a, h+a]$ ó $x \in [h-b, h+b]$ según el caso. El código es el siguiente:

```
1  #Elipse
2  elipse <- function(h, k, a, b, horizontal){
3    if (a > b){ # a tiene que ser mayor que b
4      c <- sqrt(a^2 - b^2) # calculamos c
5      if (horizontal){ # si es una elipse horizontal
6        x <- seq(h - a, h + a, 0.01) #definimos el dominio
7        ypositiva <- k + sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte positiva
8        ynegativa <- k - sqrt((b^2 - (b^2/a^2) * ((x - h)^2))) # parte negativa
9        # graficamos primero la parte positiva
10       plot(x, ypositiva, type = "l", xlim = c(h - (a + 1), h + (a + 1)), ylim = c(k - (b + 1),
             k + (b + 1)),
11            xlab = "Eje X", ylab = "Eje Y")
12       lines(x, ynegativa, type = "l") # agregamos la parte negativa
13       abline(h = 0, v = 0) # ejes coordenados
14       points(x = c(h - c, h + c), y = c(k, k), col = "red") # focos
15     } else{
16       x <- seq(h - b, h + b, 0.01)
17       ypositiva <- k + sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
18       ynegativa <- k - sqrt((a^2 - (a^2/b^2) * ((x - h)^2)))
19       plot(x, ypositiva, type = "l", xlim = c(h - (b + 1), h + (b + 1)), ylim = c(k - (a + 1),
             k + (a + 1)),
20            xlab = "Eje X", ylab = "Eje Y")
21       lines(x, ynegativa, type = "l")
22       abline(h = 0, v = 0)
23       points(x = c(h, h), y = c(k - c, k + c), col = "red")
24     }
25   } else {
26     return(print("No cumple las condiciones para ser una elipse. (a no es mayor que b)"))
27   }
28 }
29
30 elipse(1, 2, 16, 9, TRUE)
```

Cuadro 5: Código actualizado en R para gráficar una elipse.

La función `elipse(h, k, a, b, horizontal)` recibe en entrada el centro de una elipse (h, k), los valores de las constantes a y b, y una variable booleana ”`horizontal`”donde si es una elipse horizontal entonces `horizontal = TRUE`. Se valida si $a > b$ y si es horizontal (lineas 2 y 4), dependiendo si es horizontal o no el dominio cambia y en los calculos de los vectores `ypositiva` y `ynegativa`, se intercambian a y b. Las lineas para gráficar son similares que en la circunferencia solo que en esta gráfica agregamos los focos con la línea `points(x = c(h - c, h + c), y = c(k, k), col = ‘‘red’’)` o `points(x = c(h, h), y = c(k - c, k + c), col = ‘‘red’’)` según el caso si es horizontal o no.

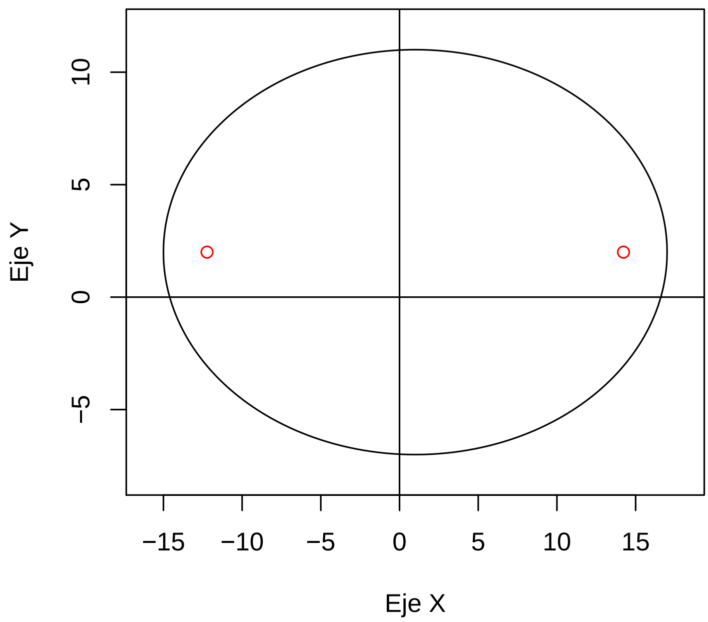

Figura 5: Grafica de una elipse horizontal con centro en (1, 2), $a = 16$ y $b = 9$.

2.5. Hipérbola

Para la hipérbola utilizaremos dos formas para graficarla. Si la hipérbola es horizontal, se utilizara la ecuación:

$$y = k \pm \sqrt{\frac{b^2}{a^2}(x-h)^2 - b^2}, \tag{7}$$

el dominio para gráficar considerado es $x \in [h-(a+3), h-a] \cup [h+a, h+(a+3)]$ y para una hipérbola vertical evaluaremos con el rango y utilizando la ecuación:

$$x = h \pm \sqrt{\frac{b^2}{a^2}(y-k)^2 - b^2}, \tag{8}$$

con rango de evaluación $y \in [k-(a+3), k-a] \cup [k+a, k+(a+3)]$.

El código del cuadro (6) crea una función `hiperbola(h, k, a, b, horizontal)` con los parámetros similares a los de la elipse, validación si es una hipérbola horizontal o no, si vamos a evaluar con dominio o rango y las lineas para gráficar idénticas a la de elipse. La figura (6) muestra una parte de la hipérbola, la otra parte de la hipérbola la veremos en la tarea.

3. Tarea

Termine de dibujar la otra parte de la hipérbola, haga dos gráficas de cada curva agregue en el código lo que ocupe para que al ejecutarse se creen todas la gráficas. Diseñe un reporte de la practica añadiendo estos dibujos con su correspondiente caption junto con la definición de cada curva, puede utilizar el libro de geometria analitica[1] para estas definiciones. El codigo guardelo en su repositorio de github. Agregue en la referencias la liga de su repositorio.[2]

```r
#Hiperbola
hiperbola <- function(h, k, a, b, horizontal){
  c <- sqrt(a^2 + b^2) # calculamos c
  if (horizontal){ # hiperbola sobre el eje x
    xizq <- seq(h - (a + 3), h - a, 0.01) # dominio izquierdo
    xder <- seq(h + a, h + (a + 3), 0.01) # dominio derecho
    yizqpositiva <- k + sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte positiva del dominio izquierdo
    yizqnegativa <- k - sqrt((b^2/a^2)*((xizq - h)^2) - b^2) # parte negativa del dominio izquierdo
    # greficamos la parte positiva del dominio izquierdo
    plot(xizq, yizqpositiva, type = "l", xlim = c(h - (a + 4), h + (a + 4)), ylim = c(k - (b + 4), k + (b + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizq, yizqnegativa, type = "l") # agregamos parte negativa del dominio izquierdo
    abline(h = 0, v = 0) # ejes coordenados
    points(x = c(h - (a + c)), y = c(k), col = "red") # focos
  } else{ # hiperbola sobre el eje y
    yizq <- seq(k - (a + 3), k - a, 0.01) # rango inferior
    yder <- seq(k + a, k + (a + 3), 0.01) # rango superior
    xizqpositiva <- h + sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte positiva del rango inferior
    xizqnegativa <- h - sqrt((b^2/a^2)*((yizq - k)^2) - b^2) # parte negativa del rango superior
    # graficamos
    plot(xizqpositiva, yizq, type = "l", xlim = c(h - (b + 4), h + (b + 4)), ylim = c(k - (a + 4), k + (a + 4)),
         xlab = "Eje X", ylab = "Eje Y")
    lines(xizqnegativa, yizq, type = "l")
    abline(h = 0, v = 0)
    points(x = c(h), y = c(k - (a + c)), col = "red") # focos
  }
}

hiperbola(1, 2, 2, 3, FALSE)
```

Cuadro 6: Código actualizado en R para gráficar una hipérbola.

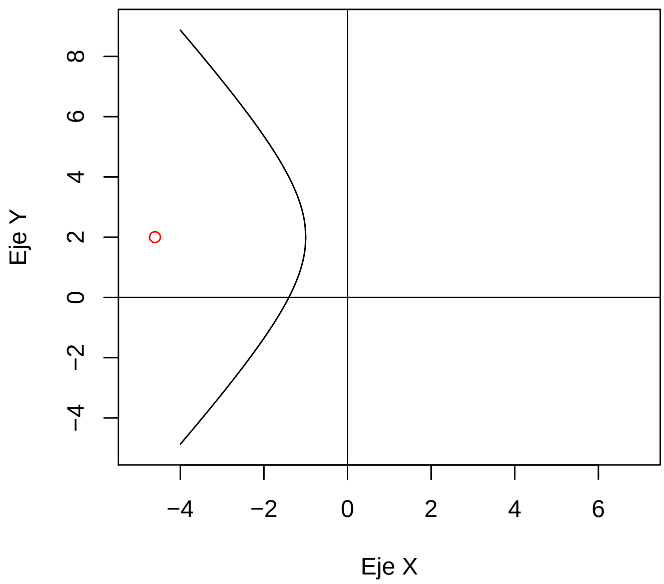

Figura 6: Grafica de una hipérbola sobre el eje X con centro en (1, 2), $a = 2$ y $b = 3$.

Referencias

[1] Charles H Lehmann. *Geometría analítica*. LIMUSA, 1965.

[2] Ángel Moreno. Repositorio de Github. `https://github.com/angelmorenos`, 2021.