Introducción a Ggplot

Análisis Numérico

Este documento fue preparado para el curso de Análisis Numérico (modalidad virtual) de la Facultad de Ciencias Económicas de la Universidad de Buenos Aires, a cargo del profesor Darío Bacchini, correspondiente a la cátedra del Prof. Dr. Javier García Fronti.

El mismo fue preparado con la colaboración de Facundo Caamaño.

Antes de empezar algunos shortcuts de utilidad:

- Alt + O colapso todo el código
- Shift + Alt + O Abro todo
- Con L en lugar de O lo hago por secciones

Primero se debe preparar el archivo

- Con el siguiente código limpio el enviroment y los gráficos de la pestaña de *plots*

```
rm(list = ls())
graphics.off()
```

- Seteo el directorio:

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

- Instalo el paquete necesario y lo cargo en la librería. Por otro lado cargamos los algoritmos que vamos a utilizar desde otra fuente:

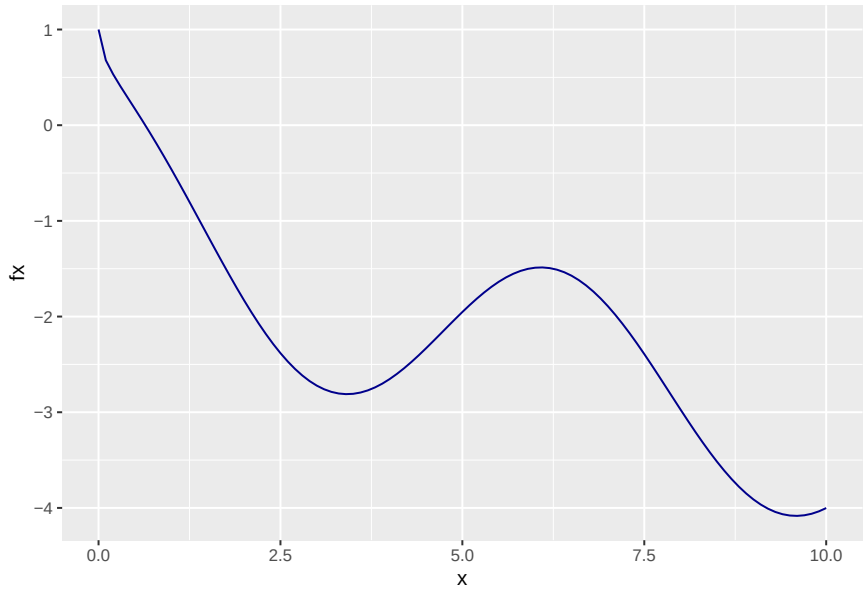
```
#install.packages("ggplot2")  Instalamos el paquete
library(ggplot2)

source('Clase Introducción a Ggplot - sourced.R')
```

Método de Punto Fijo

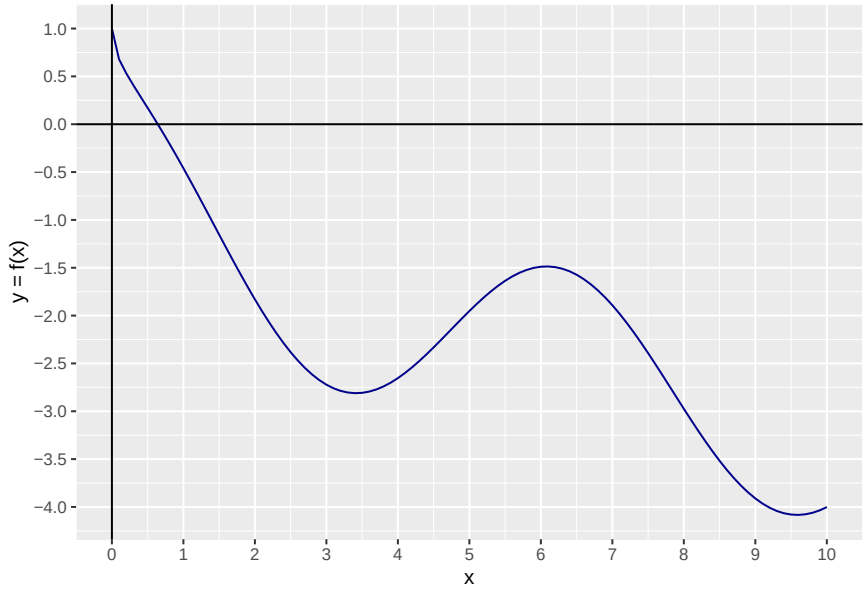
Para este ejemplo nuestra función es la siguiente: $f(x) = cos(x) - raíz(x)$

Se debe graficar $f(x)$ para identificar visualmente dónde se encuentra la raíz

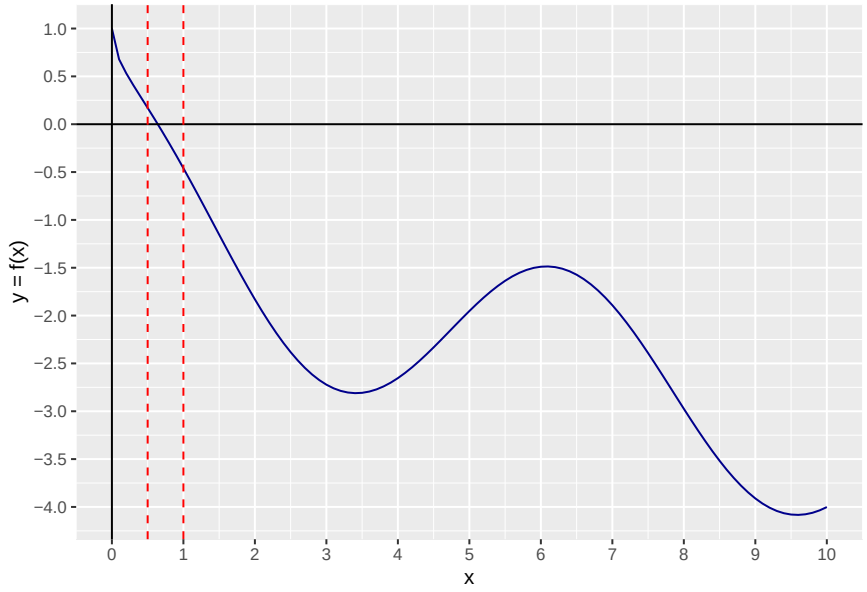
```
f = function(x){
  cos(x)-sqrt(x)
}
x <- seq(0,10,by = 0.1) #Generamos un vector "x" para crear los puntos en F(x)
fx <- f(x) #Creamos los valores de f(x)
df <- data.frame(x, fx) # Creo data frame

gg_fx = ggplot(data = df) #Cargo los datos
gg_fx = gg_fx + aes(x=x,y=fx) #Agrego capa estética (culummas de "df")
gg_fx = gg_fx + geom_line(linetype=1,colour="darkblue") #Agrego la geometria
gg_fx # Recién aquí aparece la curva
```



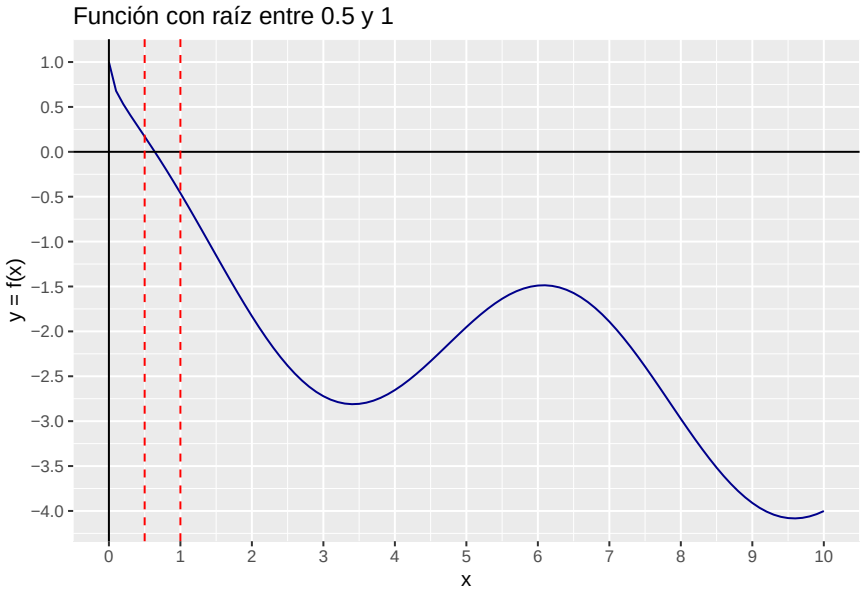

```
gg_fx = gg_fx + geom_hline(yintercept = 0,linetype=1) #Agrego linea que cruza en y=0
gg_fx = gg_fx + geom_vline(xintercept = 0,linetype=1) #Agrego linea que cruza en x=0
gg_fx = gg_fx + scale_x_continuous(name = "x", breaks = seq(0,10, by = 1)) # Cambio ticks
#en eje X
gg_fx = gg_fx + scale_y_continuous(name = "y = f(x)", breaks = seq(-4.5,1, by = 0.5))
# Cambio ticks en eje Y
gg_fx
```




```
#La raíz se encuentra entre 0.5 y 1 aproximadamente
gg_fx = gg_fx + geom_vline(xintercept = c(0.5,1),linetype=2, col = "red")
gg_fx
```




```
gg_fx = gg_fx + ggtitle("Función con raíz entre 0.5 y 1") # Agrego título
gg_fx
```



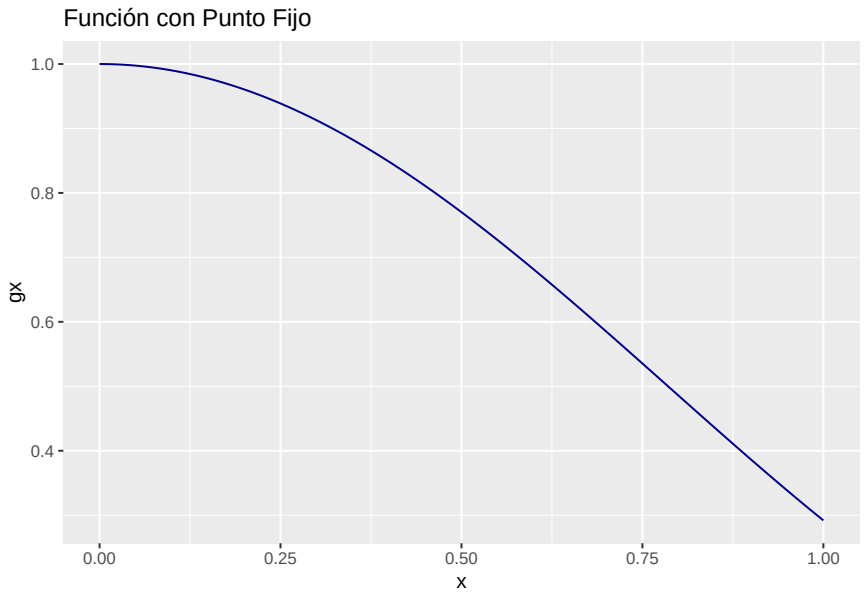
Podemos optimizar el código y graficar todo junto utilizando un signo "+" al final de cada linea:

```
gg_fx = ggplot(data = df) + #Cargo los datos
  aes(x=x,y=fx) + #Agrego capa estética (culummas de "df")
  geom_line(linetype=1,colour="darkblue") + #Agrego la geometria
  geom_hline(yintercept = 0,linetype=1) + #Agrego linea que cruza en y=0
  geom_vline(xintercept = 0,linetype=1) + #Agrego linea que cruza en x=0
  scale_x_continuous(name = "x", breaks = seq(0,10, by = 1)) + # Cambio ticks en eje X
  scale_y_continuous(name = "y = f(x)", breaks = seq(-4.5,1, by = 0.5)) + # Cambio ticks en eje Y
  geom_vline(xintercept = c(0.5,1),linetype=2, col = "red") + # Intervalo de las raíces
  ggtitle("Función con raíz entre 0.5 y 1") # Agrego título
```

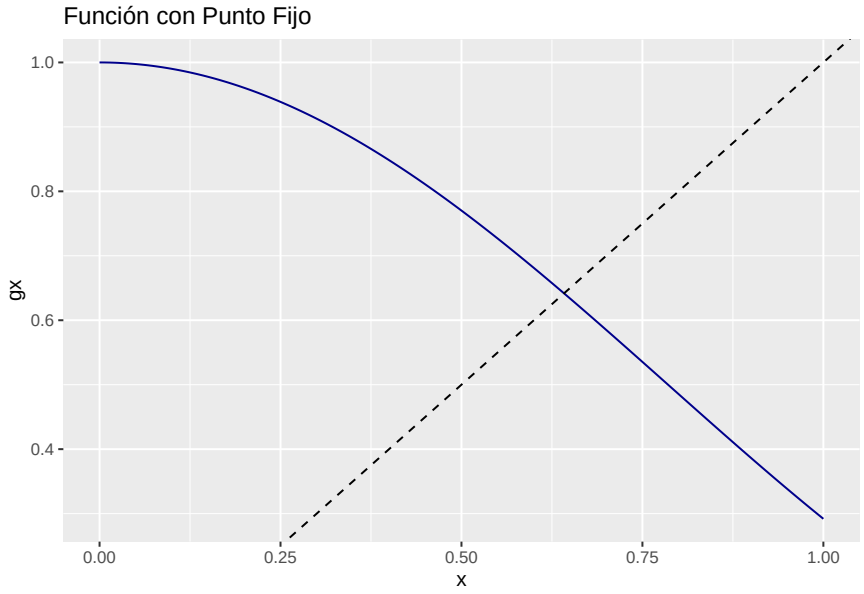
Graficamos g(x) para identificar donde hay un Punto Fijo

Si g(x) tiene punto fijo en p entonces $f(x) = x - g(x)$ tiene un cero en p Tambien puedo sacar g(x) despejando cualquiera de las x de f(x) Utilizaremos como g(x) la siguiente función $f(x) = cos(x) - raíz(x) = 0 => cos(x)^2 = x = g(x)$

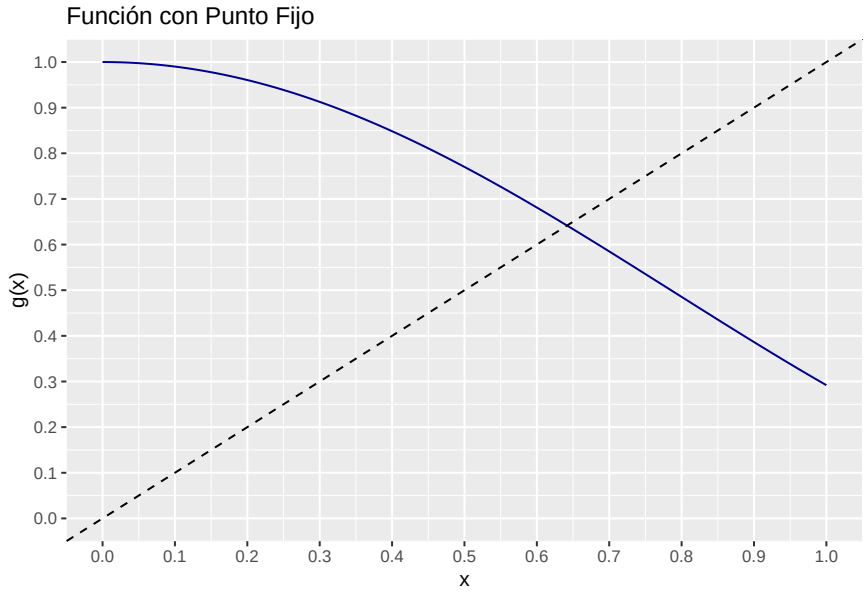
```
g = function(x){
  cos(x)^2
}
x2 = seq(0,1,by = 0.001) #Creamos un Vector para obtener puntos de la g(x)
gx = g(x2)
df2 = data.frame(x=x2, gx)

gg_gx = ggplot(data = df2) +
  aes(x=x,y=gx) +
  geom_line(linetype=1,colour="darkblue") +
  ggtitle("Función con Punto Fijo")
gg_gx
```




```
#Agrego recta x=y para ver donde cruza a g(x)
gg_gx = gg_gx +
  geom_abline(intercept = 0,slope=1,linetype=2)
gg_gx
```




```
# Corrijo escalas para que se vea el "mapeo"
gg_gx = gg_gx +
  scale_x_continuous(name = "x", limits = c(0,1), breaks = seq(0,1, by = 0.1))+
  scale_y_continuous(name = "g(x)", limits = c(0,1), breaks = seq(0,1, by = 0.1))
gg_gx
```



Gracias al gráfico, identificamos que el Punto Fijo se encuentra entre 0.6 y 0.7. Utilizamos el algoritmo para corroborar esto y obtener el resultado

```
PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300)
```

```
## [1] 0.6417149
```

```
g(PuntoFijo(g,0.6,10^-6,300))  # Chequeo que g(p)=p
```

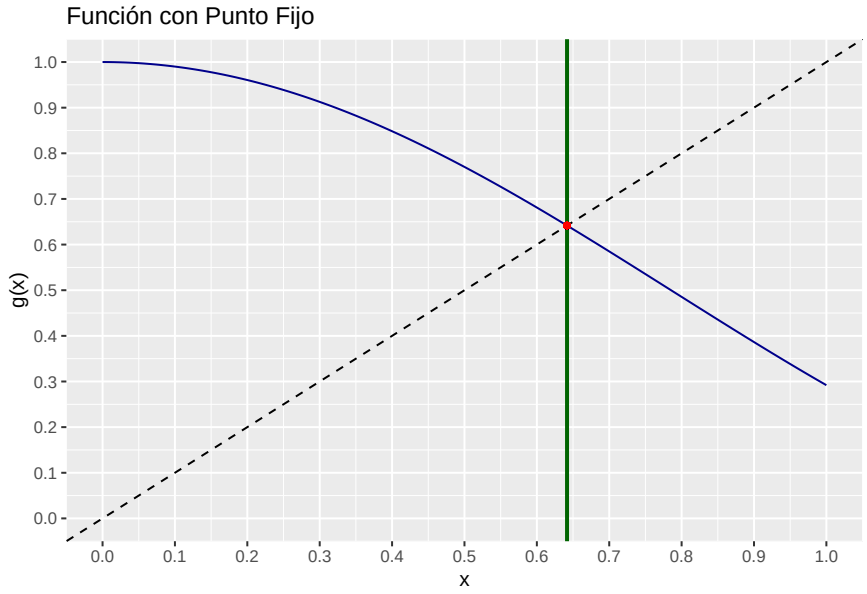
```
## [1] 0.6417139
```

```
f(PuntoFijo(g,0.6,10^-6,300))  # Chequeo que f(p)=0
```

```
## [1] -5.888511e-07
```

Verificamos gráficamente (Agrego línea vertical verde y punto rojo)

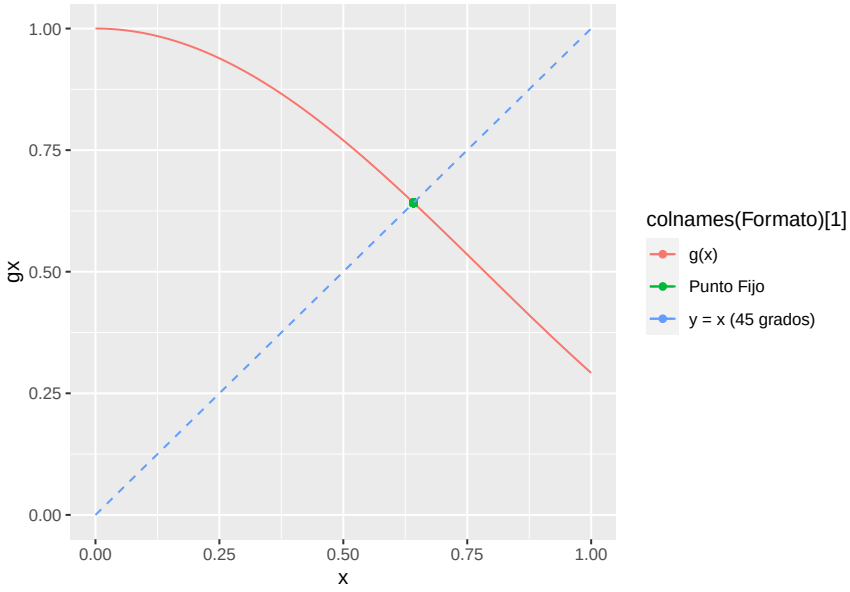
```
gg_gx = gg_gx +
  geom_vline(xintercept=PuntoFijo(g,0.6,10^-6,300),linetype=1,size=1,colour="darkgreen")+
  geom_point(aes(x =   PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300),
                 y = g(PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300))),
             pch = 16, col = "red")
gg_gx
```



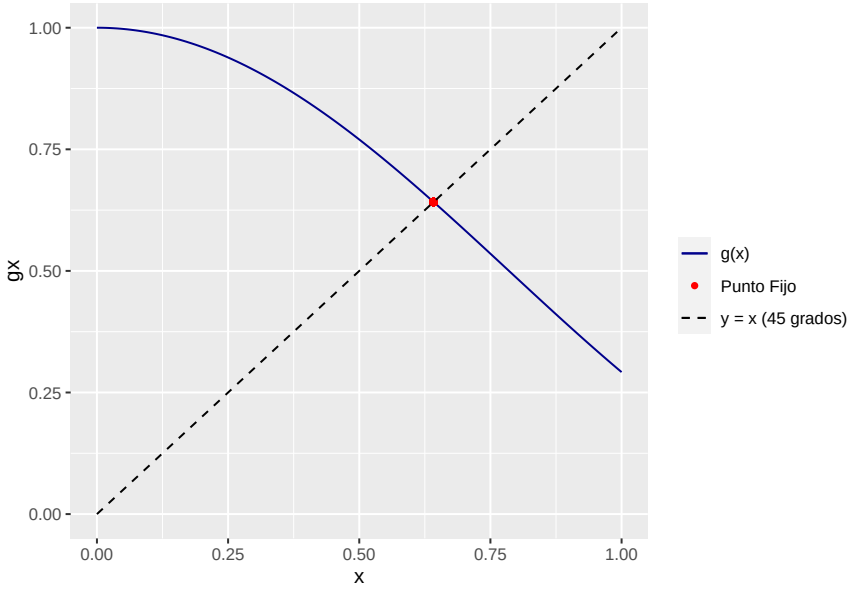

```
# Podemos hacer todo junto
gg_gx = ggplot(data = df2) +
  aes(x=x,y=gx) +
  geom_line(linetype=1,colour="darkblue") +
  ggtitle("Función con Punto Fijo") +
  geom_abline(intercept = 0,slope=1,linetype=2) +
  scale_x_continuous(name = "x", limits = c(0,1), breaks = seq(0,1, by = 0.1))+
  scale_y_continuous(name = "g(x)", limits = c(0,1), breaks = seq(0,1, by = 0.1))+
  geom_vline(xintercept=PuntoFijo(g,0.6,10^-6,300),linetype=1,size=1,colour="darkgreen")+
  geom_point(aes(x =   PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300),
                 y = g(PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300))),
             pch = 16, col = "red")
```

Agrego Leyenda manualmente al gráfico de g(x)

```
# Paso 1: Agrego columnas al data frame, repitiendo los valores del punto fijo "p" y g(p)
  df2$p0 = PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300)
  df2$gp0 = g(PuntoFijo(g, p0 = 0.6, tol = 10^-6, N = 300))
# Paso 2: defino colores, tipo de líneas y tipo de punto para cada variable
  Formato = matrix(c("darkblue", "solid", NA,
                     "red","blank",16,
                     "black", "dashed", NA), nrow = 3)
  colnames(Formato) = c("g(x)","Punto Fijo","y = x (45 grados)")
  rownames(Formato) = c("color", "linea", "punto")
# Paso 3: al graficar, dentro de "aes" agrego el "colour"
gg_gx =
  ggplot(data = df2) + # Cargo datos
    geom_line(aes(x = x, y = gx, colour = colnames(Formato)[1])) + # Cargo "g" con un "color"
    geom_point(aes(x = p0, y = gp0, colour = colnames(Formato)[2])) + # Cargo "p" con un "color"
    geom_line(aes(x = x, y = x, colour = colnames(Formato)[3]), linetype = "dashed")
gg_gx
```

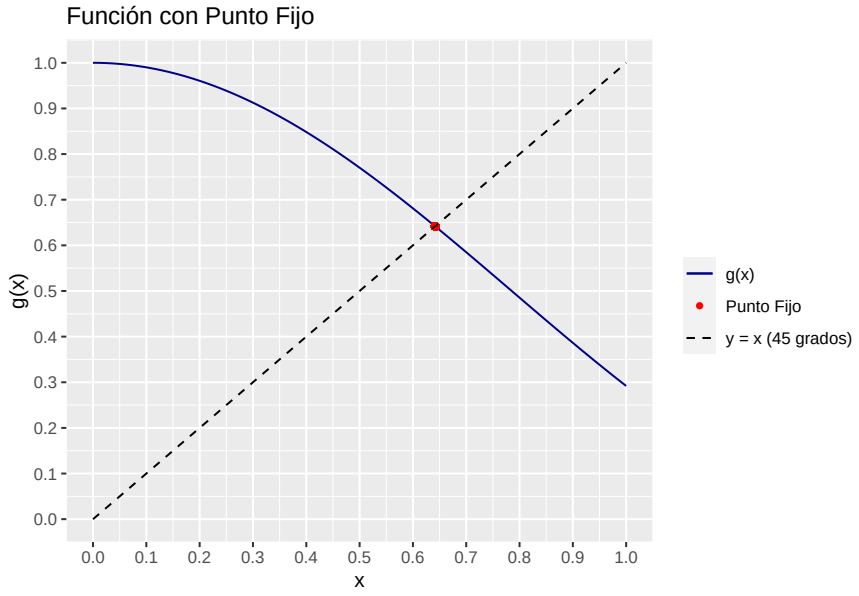



```
# Paso 4: redefino colores y tipografía de acuerdo a la matruiz definida
gg_gx = gg_gx +
    scale_color_manual(name=" ",
                       values = Formato[1,],
                       guide = guide_legend(override.aes = list(
                         colour = Formato[1,],
                         linetype = Formato[2,],
                         shape = as.numeric(Formato[3,]))))
gg_gx
```

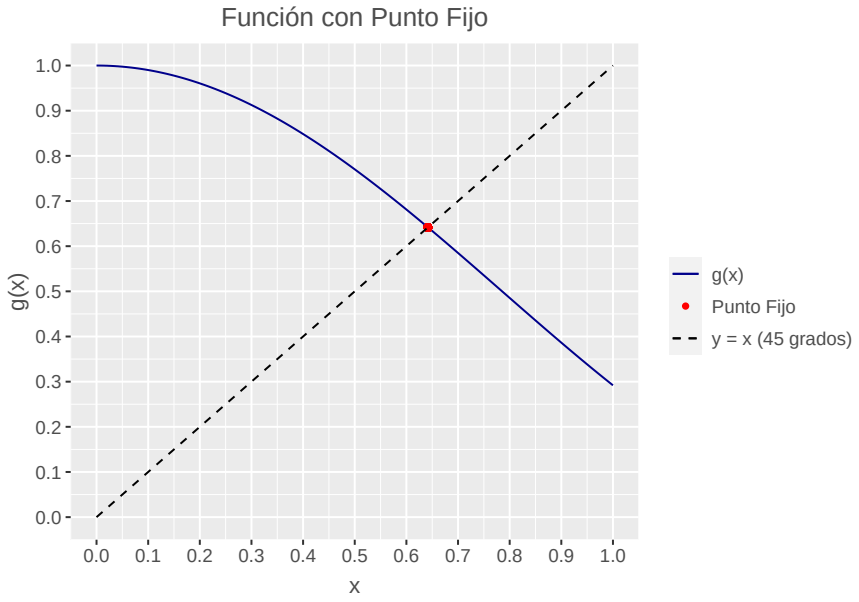



```
# Hago todo el gráfico en un solo paso:
gg_gx =
  ggplot(data = df2) +
```

```
  geom_line(aes(x = x, y = gx, colour = colnames(Formato)[1])) +
  geom_point(aes(x = p0, y = gp0, colour = colnames(Formato)[2])) +
  geom_line(aes(x = x, y = x, colour = colnames(Formato)[3]), linetype = "dashed") +
  scale_color_manual(" ", values = Formato[1,],
                     guide = guide_legend(override.aes = list(
                       colour = Formato[1,],
                       linetype = Formato[2,],
                       shape = as.numeric(Formato[3,]))))+
  ggtitle("Función con Punto Fijo") +
  scale_x_continuous(name = "x", limits = c(0,1), breaks = seq(0,1, by = 0.1))+
  scale_y_continuous(name = "g(x)", limits = c(0,1), breaks = seq(0,1, by = 0.1))
gg_gx
```




```
#Podemos darle formato al texto de los titulos y de los ejes con la funcion "theme":
gg_gx = gg_gx +
theme(plot.title=element_text(color="gray30", size=14,vjust=1.25,hjust = 0.5),
      axis.title.x=element_text(size=12,color="gray30", vjust=0),
      axis.title.y=element_text(size=12,color="gray30", vjust=1.25),
      axis.text.x=element_text(size=10,color="gray30"),
      axis.text.y=element_text(size=10,color="gray30"),
      legend.text = element_text(size=10,color="gray30"))
gg_gx
```

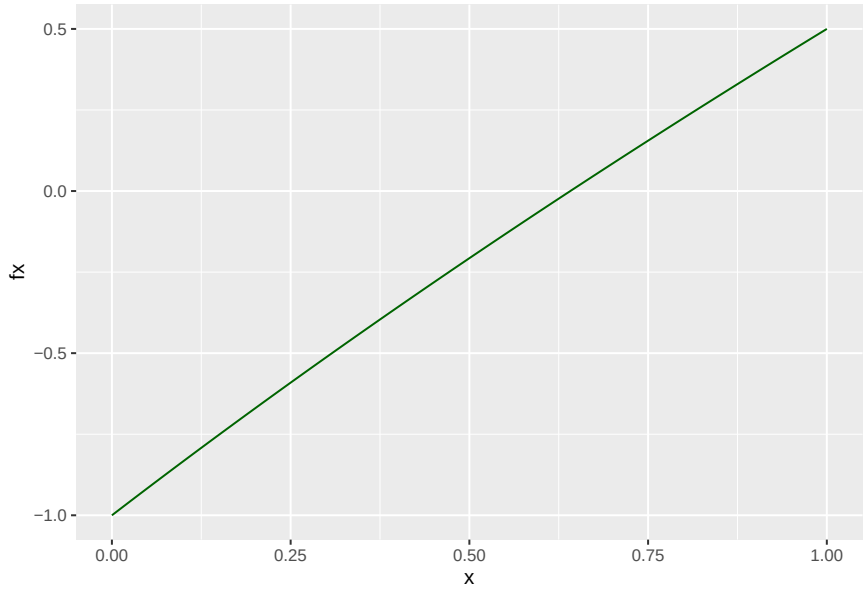

Función con Punto Fijo
g(x)
x
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
g(x)
Punto Fijo
y = x (45 grados)

Bisección

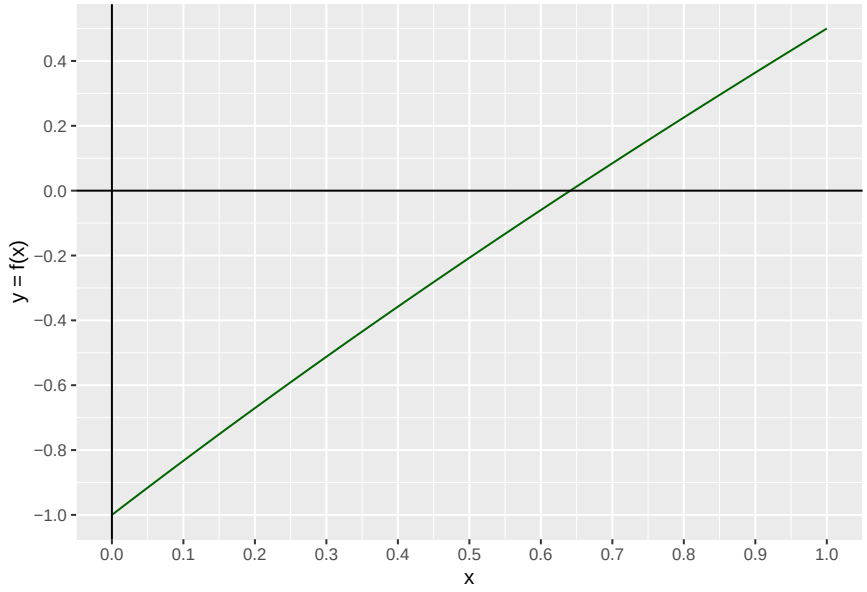
```
rm(list = ls())
graphics.off()
source('Clase Introducción a Ggplot - sourced.R')


f<- function(x){x-2^-x}
x<- seq(0,1,by=0.001)

fx<-f(x)

df<- data.frame(x,fx)  #Creo el data frame con los datos
```

Graficamos

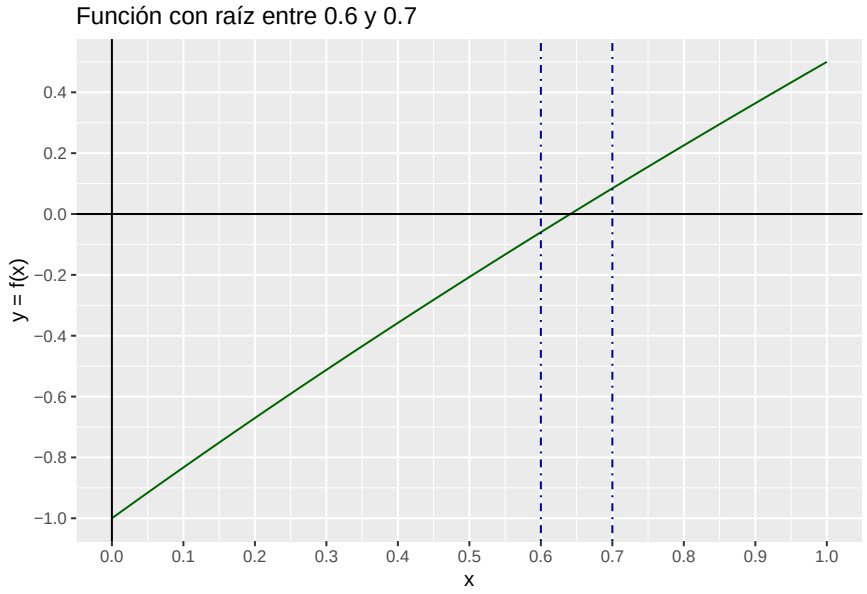
```
gg_fx = ggplot(data = df) + aes(x=x,y=fx) #Inserto data en el ggplot y creo la estética
                          #Aún no se grafica la curva, me falta la geométrica

gg_fx = ggplot(data = df) + aes(x=x,y=fx) + geom_line(linetype=1,colour="darkgreen")
gg_fx     #Con la geométrica ya grafica la curva
```




```
gg_fx = ggplot(data = df) + aes(x=x,y=fx) + geom_line(linetype=1,colour="darkgreen")+
        geom_hline(yintercept = 0,linetype=1) +
        geom_vline(xintercept = 0,linetype=1)
        #Grafico curvas en x=0 e y=0, para visualizar los ejes

gg_fx = gg_fx + scale_x_continuous(name = "x", breaks = seq(0,1, by = 0.1)) +
        scale_y_continuous(name = "y = f(x)", breaks = seq(-1,0.5, by = 0.2))
gg_fx #Cambio la escala de los ejes
```




```
#La raíz se encuentra entre 0.6 y 0.7 aproximadamente
gg_fx = gg_fx + geom_vline(xintercept = c(0.6,0.7),linetype="dotdash", col = "darkblue")
gg_fx = gg_fx + ggtitle("Función con raíz entre 0.6 y 0.7") # Agrego título
gg_fx
```

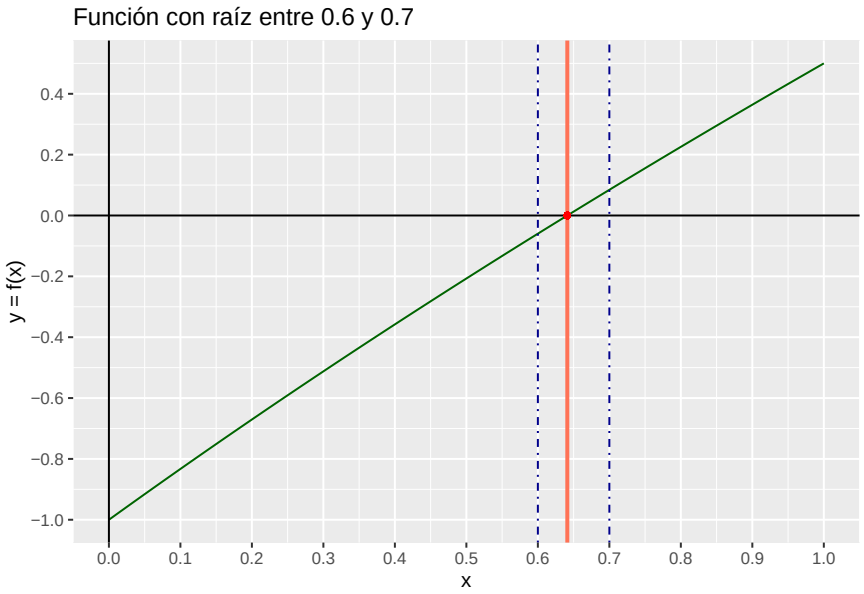


Sacamos la raíz a través del método de bisección

```
Biseccion(f=f,a=0.6,b=0.7,Tol=10^-6,N=300)
```

```
## [1] 0.6411858
```

```r
# Verificamos gráficamente
gg_fx = gg_fx +
  geom_vline(xintercept=Biseccion(f,0.6,0.7,10^-6,300),linetype=1,size=1,colour="coral1")+
  geom_point(aes(x =   Biseccion(f,0.6,0.7,10^-6,300),
                 y = 0),
             pch = 16, col = "red")
gg_fx  #Agrego el punto en la raíz y una curva en x=Raíz
```



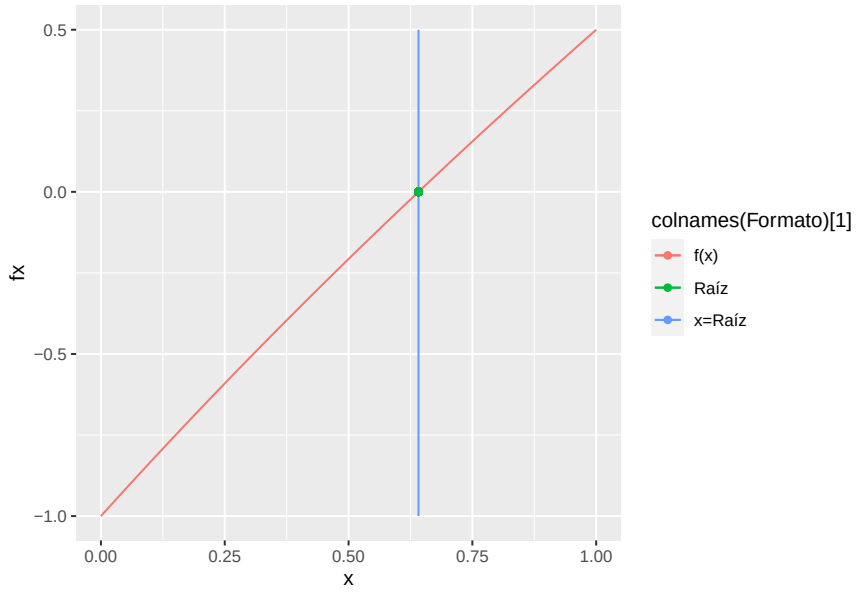
Agregamos una leyenda

```r
#Paso1_____________________________________________________________________

df$p0 = Biseccion(f,0.6,0.7,10^-6,300)
df$fp0 = f(Biseccion(f,0.6,0.7,10^-6,300)) #Puedo poner directamente 0.
#Esta columna me sirve para graficar el punto

df$vline= seq(-1,0.5,length.out = 1001)     #Para la curva en x=Raíz creo una
#columna de -1 a 0.5 para ocupar todo el grafico, y que tenga 1001 valores al
#igual que la tabla Formato.

#Paso 2_____________________________________________________________________

Formato=matrix(c("darkgreen",1,NA,
                 "coral1",1,NA,
                 "red",0,16),nrow = 3)    #Hago una tabla de formatos para cada
#curva (Color,linea,pto)

colnames(Formato)<-c("f(x)","x=Raíz","Raíz")
row.names(Formato)<-c("Color","Linea","Punto")
Formato
```

```
##       f(x)        x=Raíz   Raíz
```

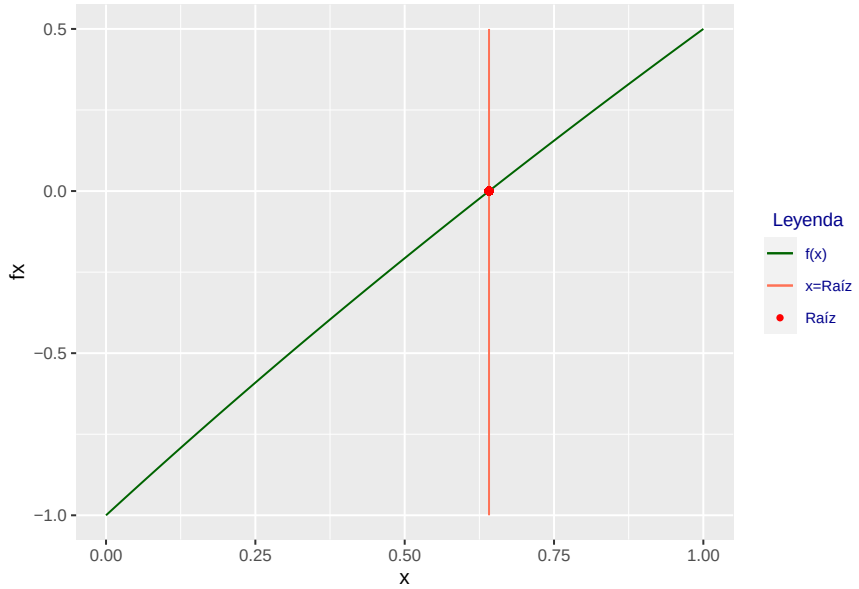
```
## Color "darkgreen" "coral1" "red"
## Linea "1"         "1"      "0"
## Punto NA          NA       "16"
```

```
#Paso 3_____________________________________________________________________

#Comienzo a graficar. La estetica debe estar dentro de la geométrica para tener
#una leyenda.

ggf<- ggplot(data = df)+                    #Asigno los datos
  geom_line(aes(x=x,y=fx,colour=colnames(Formato)[1]))+     #Asigno la primer
  #curva y le cargo un color con aes
  geom_line(aes(x=p0,y=vline,colour=colnames(Formato)[2]))+ #Asigno la segunda
  #curva y le cargo un color con aes
  geom_point(aes(x=p0,y=fp0,colour=colnames(Formato)[3]))    #Asigno el punto y
#le cargo un color con aes

ggf
```




```
# Paso 4_____________________________________________________________________
ggf<- ggplot(data = df)+
  geom_line(aes(x=x,y=fx,colour=colnames(Formato)[1]))+
  geom_line(aes(x=p0,y=vline,colour=colnames(Formato)[2]))+
  geom_point(aes(x=p0,y=fp0,colour=colnames(Formato)[3]))+
  scale_color_manual(name="Leyenda", values = Formato[1,],  #Configuro la l
                     #eyenda
                     labels= c("f(x)","x=Raíz","Raíz"),
                     guide =  guide_legend(override.aes = list(
      #override.aes es una lista especificando los parametros esteticos
                                colour = Formato[1,],
                                linetype = as.numeric(Formato[2,]),
                                shape = as.numeric(Formato[3,])),
                              title.theme = element_text(size = 10,color="blue4",
```

```
                                        hjust = 0.5),  #Titulo de la leyenda
                              label.theme = element_text(size = 8,color="blue4")))
                               #Texto de la leyenda

ggf
```




```
##Graficamos todo junto-------------------------------------------------------------

ggf<- ggplot(data = df)+                    #Asigno los datos
      geom_hline(yintercept = 0,linetype=1,colour="blue4") +
      #Marco el eje x en y=0
      geom_vline(xintercept = 0,linetype=1,colour="blue4") +
      #Marco el eje y en x=0
      geom_line(aes(x=x,y=fx,colour=colnames(Formato)[1]))+
      #Asigno la primer curva y le cargo un color con aes
      geom_line(aes(x=p0,y=vline,colour=colnames(Formato)[2]))+
      #Asigno la segunda curva y le cargo un color con aes
      geom_point(aes(x=p0,y=fp0,colour=colnames(Formato)[3]))+
      #Asigno el punto y le cargo un color con aes
      scale_color_manual(name="Leyenda", values = Formato[1,],
                         #Configuro la leyenda
                         labels= c("f(x)","x=Raíz","Raíz"),
                         guide = guide_legend(override.aes = list(
            #override.aes es una lista especificando los parametros esteticos
                              colour = Formato[1,],
                              linetype = as.numeric(Formato[2,]),
                              shape = as.numeric(Formato[3,])),
                              title.theme = element_text(size = 10,
                                                         color="blue4",
                                                         hjust = 0.5),
                                  label.theme = element_text(size = 8,
                                                             color="blue4")))+
      ggtitle("Raíz por Método de Bisección")+    #Agrego titulo
```

```
      scale_y_continuous(name = "y = f(x)", breaks = seq(-1,0.5, by = 0.25,),
                         expand=c(0,0))+   #Configuro eje y
      scale_x_continuous(name = "x", breaks = seq(0,1, by = 0.1),expand=c(0,0))+
      #Configuro eje x

      theme(plot.title=element_text(color="blue4", size=12,
                                    vjust=0.5,hjust = 0.5),
            #Configuro los colores del texto del gráfico
            axis.title.x=element_text(size=10,color="blue4", vjust=0),
            #Formato nombre del eje x
            axis.title.y=element_text(size=10,color="blue4", vjust=1.25),
            #Formato nombre del eje y
            axis.text.x=element_text(size=8,color="blue4"),
            #Formato numeros del eje x
            axis.text.y=element_text(size=8,color="blue4"),
            #Formato numeros del eje y
            legend.background = element_rect(colour = "darkblue"))
#Agrego recuadro a la leyenda

#Con el argumento expand configurado en 0,0 quito los vacios de data a
#los costados del gráfico

ggf
```