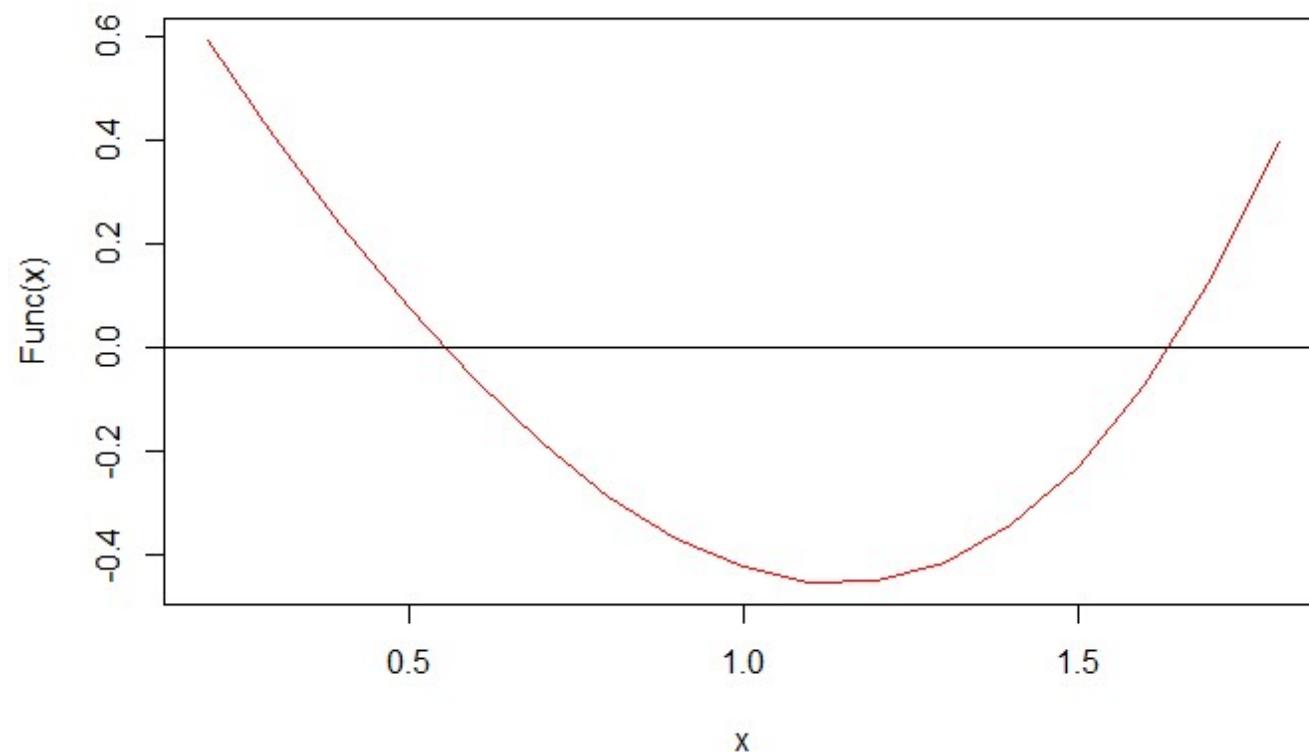


# Tarea Miercoles 6 de Febrero del 2019

[Code ▾](#)

Se graficaron tres métodos numéricos para encontrar raíces: Bisección, Punto Fijo y Secante a partir de esto se analiza cual de los tres es la mejor opción en cuestión a sus iteraciones.

## Grafica



## Metodo por Biseccion

[Hide](#)

```
Func = function(x) {exp(x) - x*pi}
fBiseccion <- function(x1,x2) {
  if(Func(x1)*Func(x2) < 0)
  {
    x = seq(x1,x2,0.1)

    plot(x,Func(x),type="l",col="red")

    abline(h=0)
    ptoM = (x1+x2)/2

    error = abs(x1-x2)/2

    i = 0

    aux = Func(ptoM)

    points(rbind(c(ptoM,aux)),pch=15,cex=0.4,col="blue")

    valor = log10((x2-x1)/1.e-8)/log10(2)

    cat("Iteracion=",i,"\tFunc(x)=",Func(ptoM),"\tX=",ptoM,"\tError=",error,"\n")

    while (error > 1.e-8) {

      i = i+1
      if (Func(ptoM) == 0) break

      if (Func(ptoM)*Func(x1) < 0) x2 = ptoM else {x1 = ptoM}

      ptoM = (x1+x2)/2

      error = abs(x1-x2)/2

      aux = Func(ptoM)

      points(rbind(c(ptoM,aux)),pch=15,cex=0.4,col="blue")
```

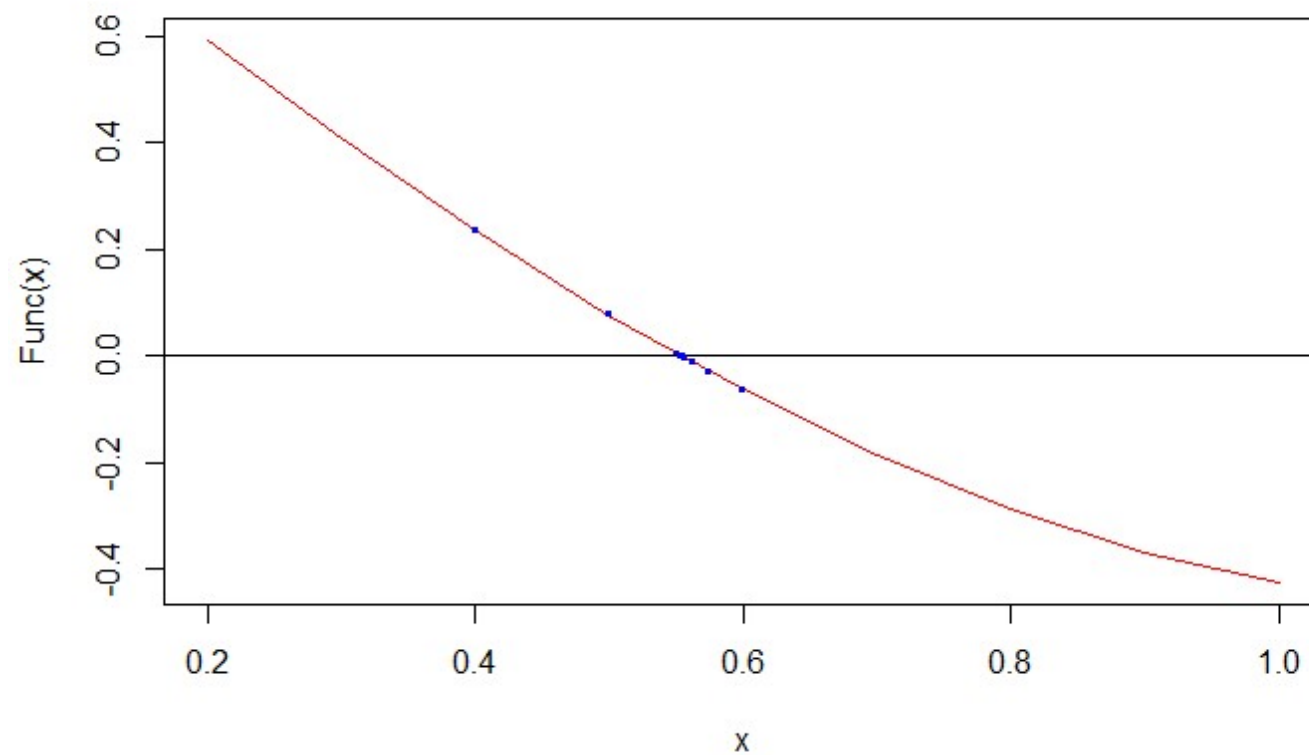
```
cat("Iteracion=", i, "\tFunc(x)=", Func(ptoM), "\tX=", ptoM, "\tError=", error, "\n")
}
cat ("Valor aproximado de iteraciones:", valor)

}

}
fBiseccion(0.2, 1)
```

Iteracion= 0	Func(x)= -0.06283679	X= 0.6	Error= 0.4
Iteracion= 1	Func(x)= 0.2351876	X= 0.4	Error= 0.2
Iteracion= 2	Func(x)= 0.07792494	X= 0.5	Error= 0.1
Iteracion= 3	Func(x)= 0.005377058	X= 0.55	Error= 0.05
Iteracion= 4	Func(x)= -0.02928525	X= 0.575	Error= 0.025
Iteracion= 5	Func(x)= -0.01209121	X= 0.5625	Error= 0.0125
Iteracion= 6	Func(x)= -0.003391141	X= 0.55625	Error= 0.00625
Iteracion= 7	Func(x)= 0.000984469	X= 0.553125	Error= 0.003125
Iteracion= 8	Func(x)= -0.001205462	X= 0.5546875	Error= 0.0015625
Iteracion= 9	Func(x)= -0.0001110274	X= 0.5539063	Error= 0.00078125
Iteracion= 10	Func(x)= 0.0004365881	X= 0.5535156	Error= 0.000390625
Iteracion= 11	Func(x)= 0.0001627472	X= 0.5537109	Error= 0.0001953125
Iteracion= 12	Func(x)= 2.585159e-05	X= 0.5538086	Error= 9.765625e-05
Iteracion= 13	Func(x)= -4.258998e-05	X= 0.5538574	Error= 4.882813e-05
Iteracion= 14	Func(x)= -8.369718e-06	X= 0.553833	Error= 2.441406e-05
Iteracion= 15	Func(x)= 8.740804e-06	X= 0.5538208	Error= 1.220703e-05
Iteracion= 16	Func(x)= 1.855109e-07	X= 0.5538269	Error= 6.103516e-06
Iteracion= 17	Func(x)= -4.092111e-06	X= 0.55383	Error= 3.051758e-06
Iteracion= 18	Func(x)= -1.953302e-06	X= 0.5538284	Error= 1.525879e-06
Iteracion= 19	Func(x)= -8.838962e-07	X= 0.5538277	Error= 7.629395e-07
Iteracion= 20	Func(x)= -3.491928e-07	X= 0.5538273	Error= 3.814697e-07
Iteracion= 21	Func(x)= -8.1841e-08	X= 0.5538271	Error= 1.907349e-07
Iteracion= 22	Func(x)= 5.183492e-08	X= 0.553827	Error= 9.536743e-08
Iteracion= 23	Func(x)= -1.500304e-08	X= 0.553827	Error= 4.768372e-08
Iteracion= 24	Func(x)= 1.841594e-08	X= 0.553827	Error= 2.384186e-08
Iteracion= 25	Func(x)= 1.70645e-09	X= 0.553827	Error= 1.192093e-08
Iteracion= 26	Func(x)= -6.648295e-09	X= 0.553827	Error= 5.960464e-09

Valor aproximado de iteraciones: 26.2535



MÃ©todo por Secante

Hide

```
Func = function(x) {exp(x) - x*pi}
fSecante <- function(x0,x1) {

  x = seq(x0,x1,0.1)

  plot(x,Func(x),type="l",col="red")

  abline(h=0)

  x2 = x1-(((x1-x0)*(Func(x1)))/(Func(x1)-Func(x0)))

  error = abs(x2-x1)/x2

  i = 0

  aux = Func(x2)

  points(rbind(c(x2,aux)),pch=15,cex=0.4,col="blue")

  cat("Iteracion=",i,"\tFunc(x)=",Func(x2),"\tX=",x2,"\tError=",error,"\n")

  while (error > 1.e-8) {

    x0 = x1

    x1 = x2

    x2 = x1-(((x1-x0)*(Func(x1)))/(Func(x1)-Func(x0)))

    i = i+1

    error = abs(x2-x1)/x2

    aux = Func(x2)

    points(rbind(c(x2,aux)),pch=15,cex=0.4,col="blue")
```

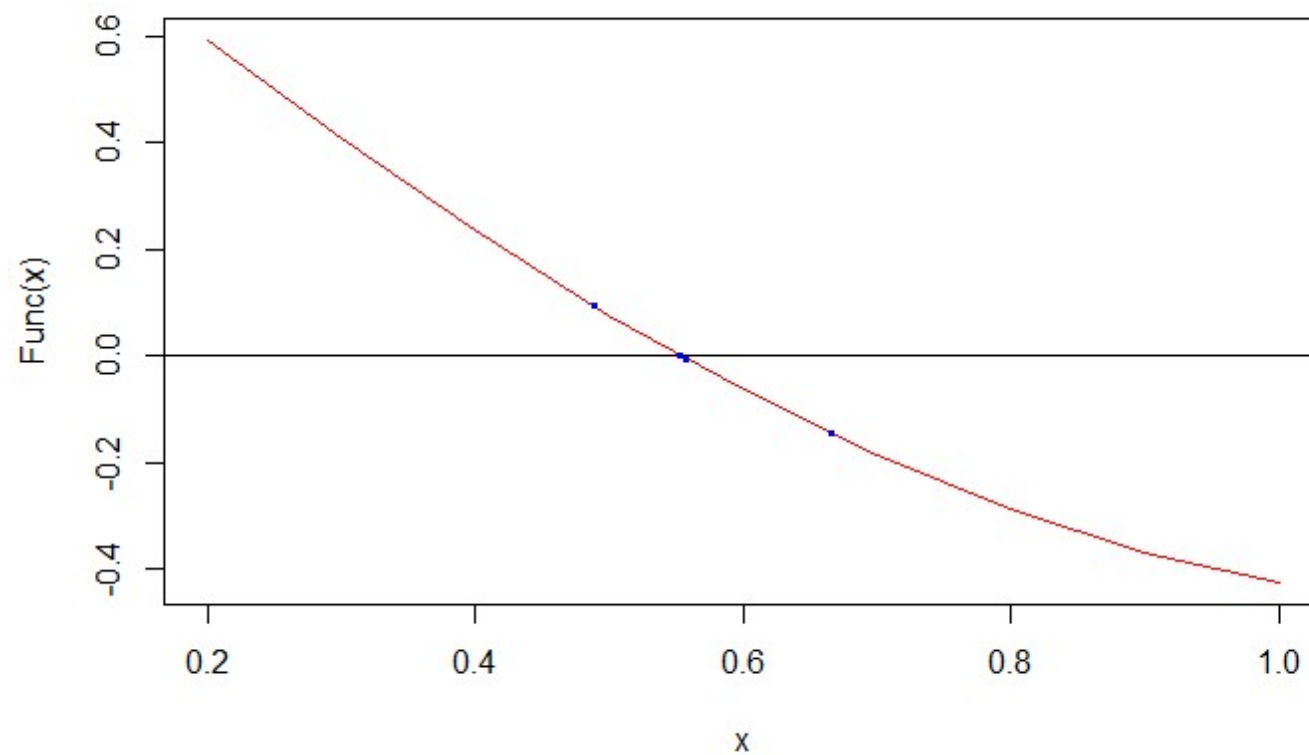
```
cat("Iteracion=", i, "\tFunc(x)=", Func(x2), "\tX=", x2, "\tError=", error, "\n")

}

}

fSecante(0.2, 1)
```

Iteracion= 0	Func(x)= -0.1468369	X= 0.6668139	Error= 0.4996687
Iteracion= 1	Func(x)= 0.09315172	X= 0.489857	Error= 0.3612421
Iteracion= 2	Func(x)= -0.006590853	X= 0.5585429	Error= 0.1229735
Iteracion= 3	Func(x)= -0.0002483718	X= 0.5540043	Error= 0.00819249
Iteracion= 4	Func(x)= 7.304429e-07	X= 0.5538265	Error= 0.0003209212
Iteracion= 5	Func(x)= -8.035195e-11	X= 0.553827	Error= 9.410368e-07
Iteracion= 6	Func(x)= 0	X= 0.553827	Error= 1.035069e-10



Método del Punto Fijo

Hide



```
library(Deriv)
Func = function(x) {exp(x) - x*pi}
FuncG = function(x) {exp(x)/pi}
fPuntoFijo = function(a,b)
{
  x = seq(a,b,0.1)
  FuncDerG = Deriv(FuncG)
  print (FuncDerG(a))
  i = 0
  if(FuncDerG(a) < 1)
  {
    fijo = FuncG(a)

    plot(x,Func(x),type = 'l', col = c("blue"))

    abline(h = 0)

    aux = Func(fijo)

    points(rbind(c(fijo,aux)),pch=15,cex=0.4,col="red")
    cat("Iteracion=",i,"\tFunc(x)=",Func(fijo),"\tX=",fijo,"\tError=---\n")
    repeat
    {
      fijoN = FuncG(fijo)

      error = abs(fijoN-fijo)/fijoN

      if(error < 1.e-8)
      {
        break

      }

      i = i + 1

      fijo = fijoN
    }
  }
}
```

```
    aux = Func(fijo)

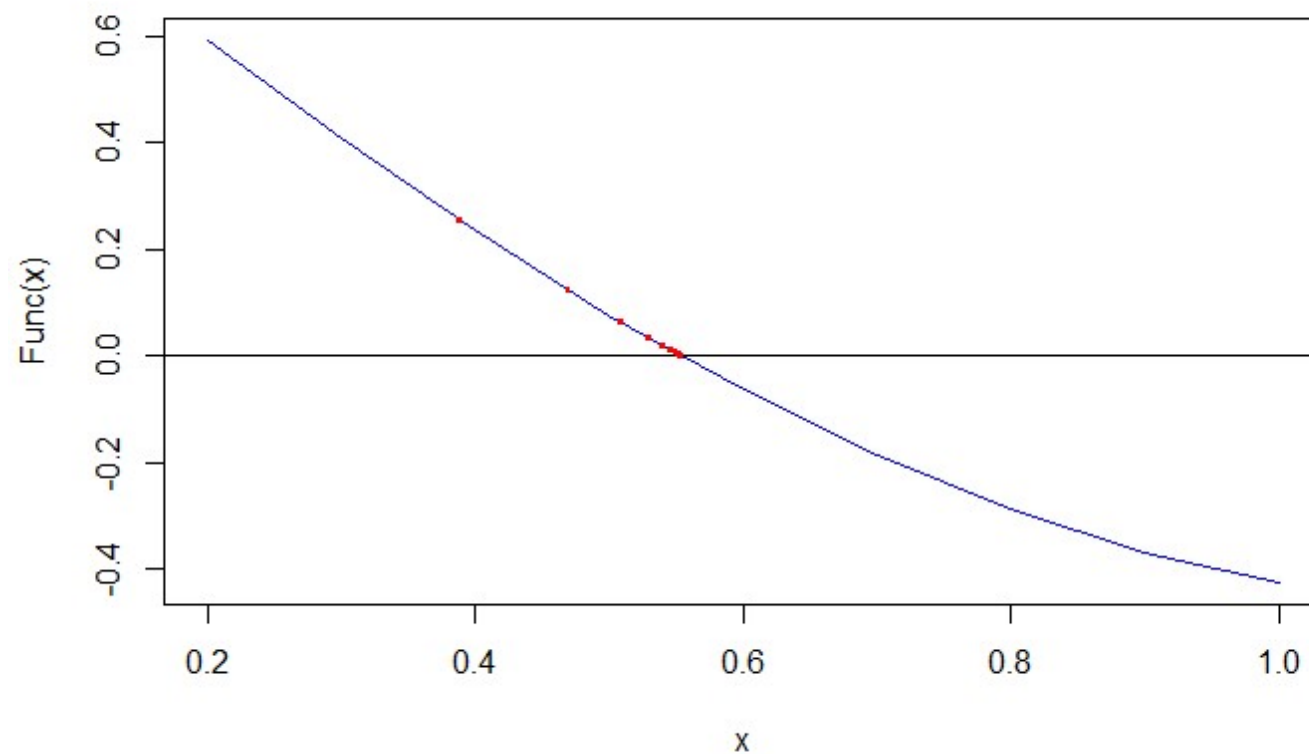
    points(rbind(c(fijo, aux)), pch=15, cex=0.4, col="red")

    cat("Iteracion=", i, "\tFunc(x)=", Func(fijo), "\tX=", fijo, "\tError=", error, "\n")
  }

}

fPuntoFijo(0.2, 1)
```

```
[1] 0.3887846
Iteracion= 0   Func(x)= 0.253784   X= 0.3887846   Error=---
Iteracion= 1   Func(x)= 0.1241141   X= 0.4695665   Error= 0.1720351
Iteracion= 2   Func(x)= 0.06444782   X= 0.5090732   Error= 0.0776052
Iteracion= 3   Func(x)= 0.03448326   X= 0.5295876   Error= 0.03873652
Iteracion= 4   Func(x)= 0.01874309   X= 0.540564   Error= 0.02030539
Iteracion= 5   Func(x)= 0.01027428   X= 0.5465301   Error= 0.01091634
Iteracion= 6   Func(x)= 0.005658051   X= 0.5498005   Error= 0.005948348
Iteracion= 7   Func(x)= 0.003123802   X= 0.5516015   Error= 0.003265063
Iteracion= 8   Func(x)= 0.001727058   X= 0.5525959   Error= 0.001799393
Iteracion= 9   Func(x)= 0.0009555773   X= 0.5531456   Error= 0.0009938428
Iteracion= 10  Func(x)= 0.0005289445   X= 0.5534498   Error= 0.0005495887
Iteracion= 11  Func(x)= 0.0002928579   X= 0.5536181   Error= 0.0003041235
Iteracion= 12  Func(x)= 0.0001621663   X= 0.5537114   Error= 0.0001683541
Iteracion= 13  Func(x)= 8.980402e-05   X= 0.553763   Error= 9.321523e-05
Iteracion= 14  Func(x)= 4.973342e-05   X= 0.5537916   Error= 5.161781e-05
Iteracion= 15  Func(x)= 2.754295e-05   X= 0.5538074   Error= 2.85851e-05
Iteracion= 16  Func(x)= 1.52538e-05   X= 0.5538162   Error= 1.583051e-05
Iteracion= 17  Func(x)= 8.447895e-06   X= 0.553821   Error= 8.767155e-06
Iteracion= 18  Func(x)= 4.678651e-06   X= 0.5538237   Error= 4.855423e-06
Iteracion= 19  Func(x)= 2.591156e-06   X= 0.5538252   Error= 2.689045e-06
Iteracion= 20  Func(x)= 1.43505e-06   X= 0.553826   Error= 1.48926e-06
Iteracion= 21  Func(x)= 7.947691e-07   X= 0.5538265   Error= 8.247904e-07
Iteracion= 22  Func(x)= 4.401644e-07   X= 0.5538267   Error= 4.567906e-07
Iteracion= 23  Func(x)= 2.437749e-07   X= 0.5538269   Error= 2.529828e-07
Iteracion= 24  Func(x)= 1.350091e-07   X= 0.5538269   Error= 1.401087e-07
Iteracion= 25  Func(x)= 7.477169e-08   X= 0.553827   Error= 7.759596e-08
Iteracion= 26  Func(x)= 4.141058e-08   X= 0.553827   Error= 4.297473e-08
Iteracion= 27  Func(x)= 2.29343e-08   X= 0.553827   Error= 2.380057e-08
Iteracion= 28  Func(x)= 1.270163e-08   X= 0.553827   Error= 1.31814e-08
```



Comparando los resultados entregados por los tres métodos, el método por secante presenta el menor número de iteraciones, seguido por el método de bisección y finalmente el punto fijo. Las tres usaron el mismo intervalo para la función.