

1.3. Applications

1. 3D plot:

Example :

Let $z = x^2 + y^2, x, y = -1, -0.99, \dots, 0, 0.01, \dots, 1$, Please write a program to plot this function.

Example (Splus):

```
x=y=seq(-1,1,by=0.01)
n=length(x)
z=matrix(0,n,n)
for(i in 1:n)
{
  for(j in 1:n)
  {
    z[i,j]=x[i]^2+y[j]^2
  }
}
persp(x,y,z)
```

2. Newton-Raphson method:

Objective: Find the solutions for **possibly nonlinear** equations

$$\begin{aligned} f_1(x) &= f_1(x_1, \dots, x_p) = 0 \\ f_2(x) &= f_2(x_1, \dots, x_p) = 0 \\ &\vdots \\ f_p(x) &= f_p(x_1, \dots, x_p) = 0 \\ \Leftrightarrow f(x) &= \begin{bmatrix} f_1(x_1, \dots, x_p) \\ f_2(x_1, \dots, x_p) \\ \vdots \\ f_p(x_1, \dots, x_p) \end{bmatrix} = 0 \end{aligned}$$

Newton-Raphson algorithm:

$$\begin{aligned} \hat{x}_{t+1} &= \begin{bmatrix} \hat{x}_{(t+1)1} \\ \hat{x}_{(t+1)2} \\ \vdots \\ \hat{x}_{(t+1)p} \end{bmatrix} = \begin{bmatrix} \hat{x}_{t1} \\ \hat{x}_{t2} \\ \vdots \\ \hat{x}_{tp} \end{bmatrix} - \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_p} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_p(x)}{\partial x_1} & \frac{\partial f_p(x)}{\partial x_2} & \dots & \frac{\partial f_p(x)}{\partial x_p} \end{bmatrix}_{x=\hat{x}_t}^{-1} \begin{bmatrix} f_1(\hat{x}_t) \\ f_2(\hat{x}_t) \\ \vdots \\ f_p(\hat{x}_t) \end{bmatrix} \\ &= \hat{x}_t - \left[\frac{\partial f(x)}{\partial x} \right]_{x=\hat{x}_t}^{-1} f(\hat{x}_t), t = 0, 1, 2, \dots \end{aligned}$$

The converge criteria are:

1. $\|\hat{x}_{N+1} - \hat{x}_N\| < \varepsilon_1$, ε_1 is some pre-specified small number.
2. $\|f(\hat{x}_N)\| < \varepsilon_2$, ε_2 is some pre-specified small number.

Example:

$f(x) = x^2 - 1$. Please find the solutions of $f(x) = 0$ by Newton-Raphson method.

Example (Splus):

```
newton1=function(initial,error)
{
  xold = initial
  repeat
  {
    f=xold^2-1
    df=2*xold
    xnew=xold-f/df
    cri1=abs(f)
    cri2=abs(xnew-xold)
    if(cri1 < error && cri2 < error) break
    xold=xnew
  }
  list(xnew)
}
solution=c(newton1(10,0.000001),newton1(-10,0.000001))
solution
```

Example:

Please write a program to find all solutions to 7 decimal places of accuracy using Newton's method $f(x) = x^3 - 4.5x^2 + 6.5x - 3 = 0$.

Example (Splus):

```
x=seq(0.5,3,by=0.05)
y=x^3-4.5*x^2+6.5*x-3
plot(x,y,type="n")
points(x,y)
abline(h=0)

newton=function(initial,error)
```

```

{
    xold=initial
    repeat
    {
        f=xold^3-4.5*xold^2+6.5*xold-3
        df=3*xold^2-9*xold+6.5
        xnew=xold-df^(-1)*f
        if((abs(f) < error) && (abs(xnew-xold) < error)) break
        xold=xnew
    }
    xnew
}
list(newton(0.9,1e-7),
      newton(1.4,1e-7),
      newton(2.1,1e-7))

```

Example:

Please write a program to find all solutions of the following equations to **5 decimal places** of accuracy using Newton-Raphson method:

$$7x^3 - 10x - y - 1 = 0$$

$$8y^3 - 11y + x - 1 = 0$$

with starting point $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, and $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ Please save the solutions as outputs in a list.

Example (Splus):

```

newton2=function(initial,error)
{
    xold = initial
    repeat
    {
        f=c(7*xold[1]^3-10*xold[1]-xold[2]-1,
            8*xold[2]^3-11*xold[2]+xold[1]-1)
        hm=matrix(c(21*xold[1]^2-10,1,-1,24*xold[2]^2-11),2,2)
        xnew=xold-solve(hm)%*%f
        cri1=sqrt(sum(f^2))
        cri2=sqrt(sum((xnew-xold)^2))
        if(cri1 < error && cri2 < error) break
        xold=xnew
    }
}

```

```

        }
        list(xnew)
    }

solution=c(newton2(c(1,0),0.00001),
           newton2(c(1,1),0.00001),
           newton2(c(1,-1),0.00001))

solution

```

Note:

Fisher's Scoring Method:

Let $l(\beta) = l(\beta_1, \dots, \beta_p) = \log[f(y|\beta)]$, where $f(y|\beta)$ is the density function.

Let

$$U(\beta) = \frac{\partial l(\beta)}{\partial \beta} = \begin{bmatrix} \frac{\partial l(\beta)}{\partial \beta_1} \\ \frac{\partial l(\beta)}{\partial \beta_2} \\ \vdots \\ \frac{\partial l(\beta)}{\partial \beta_p} \end{bmatrix} = \begin{bmatrix} U_1(\beta) \\ U_2(\beta) \\ \vdots \\ U_p(\beta) \end{bmatrix}$$

and

$$\begin{aligned} A(\beta) &= -\frac{\partial^2 l(\beta)}{\partial \beta^t \partial \beta} = -\frac{\partial U(\beta)}{\partial \beta} \\ &= \begin{bmatrix} -\frac{\partial^2 l(\beta)}{\partial \beta_1^2} & -\frac{\partial^2 l(\beta)}{\partial \beta_1 \partial \beta_2} & \cdots & -\frac{\partial^2 l(\beta)}{\partial \beta_1 \partial \beta_p} \\ -\frac{\partial^2 l(\beta)}{\partial \beta_2 \partial \beta_1} & -\frac{\partial^2 l(\beta)}{\partial \beta_2^2} & \cdots & -\frac{\partial^2 l(\beta)}{\partial \beta_2 \partial \beta_p} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\partial^2 l(\beta)}{\partial \beta_p \partial \beta_1} & -\frac{\partial^2 l(\beta)}{\partial \beta_p \partial \beta_2} & \cdots & -\frac{\partial^2 l(\beta)}{\partial \beta_p^2} \end{bmatrix} \\ &= \begin{bmatrix} A_{11}(\beta) & A_{12}(\beta) & \cdots & A_{1p}(\beta) \\ A_{21}(\beta) & A_{22}(\beta) & \cdots & A_{2p}(\beta) \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1}(\beta) & A_{p2}(\beta) & \cdots & A_{pp}(\beta) \end{bmatrix}. \end{aligned}$$

$U(\beta)$ is called the score function while

$$I(\beta) = \begin{bmatrix} I_{11}(\beta) & I_{12}(\beta) & \cdots & I_{1p}(\beta) \\ I_{21}(\beta) & I_{22}(\beta) & \cdots & I_{2p}(\beta) \\ \vdots & \vdots & \ddots & \vdots \\ I_{p1}(\beta) & I_{p2}(\beta) & \cdots & I_{pp}(\beta) \end{bmatrix} = E[A(\beta)] = E\left[-\frac{\partial^2 l(\beta)}{\partial \beta^t \partial \beta}\right]$$

is called the information matrix.

If $\hat{\beta}$ is the maximum likelihood estimate (MLE), then $U(\hat{\beta}) = \mathbf{0}..$

Let $\hat{\beta}_t = \begin{bmatrix} \hat{\beta}_{t1} \\ \hat{\beta}_{t2} \\ \vdots \\ \hat{\beta}_{tp} \end{bmatrix}$ and $\hat{\beta}_{t+1} = \begin{bmatrix} \hat{\beta}_{(t+1)1} \\ \hat{\beta}_{(t+1)2} \\ \vdots \\ \hat{\beta}_{(t+1)p} \end{bmatrix}$ be the maximum likelihood estimate at the t^{th} and $(t + 1)^{th}$ iterations, respectively.

1. **Newton-Raphson method:**

$$\hat{\beta}_{t+1} = \hat{\beta}_t + A^{-1}(\hat{\beta}_t)U(\hat{\beta}_t), t = 0, 1, 2, \dots$$

2. **Fisher's scoring method:**

$$\hat{\beta}_{t+1} = \hat{\beta}_t + I^{-1}(\hat{\beta}_t)U(\hat{\beta}_t), t = 0, 1, 2, \dots$$

The converge criteria are:

1. $\|\hat{\beta}_{N+1} - \hat{\beta}_N\| < \varepsilon_1$, ε_1 is some pre-specified small number.
2. $\|U(\hat{\beta}_N)\| < \varepsilon_2$, ε_2 is some pre-specified small number.

Useful links:

Wikipedia : http://en.wikipedia.org/wiki/Main_Page

Taiwan R User Group : <https://www.facebook.com/Tw.R.User>

R 語言翻轉教室:

<http://wush978.github.io/DataScienceAndR/>