LAB10

Clayton Glenn

April 2, 2018

Task1

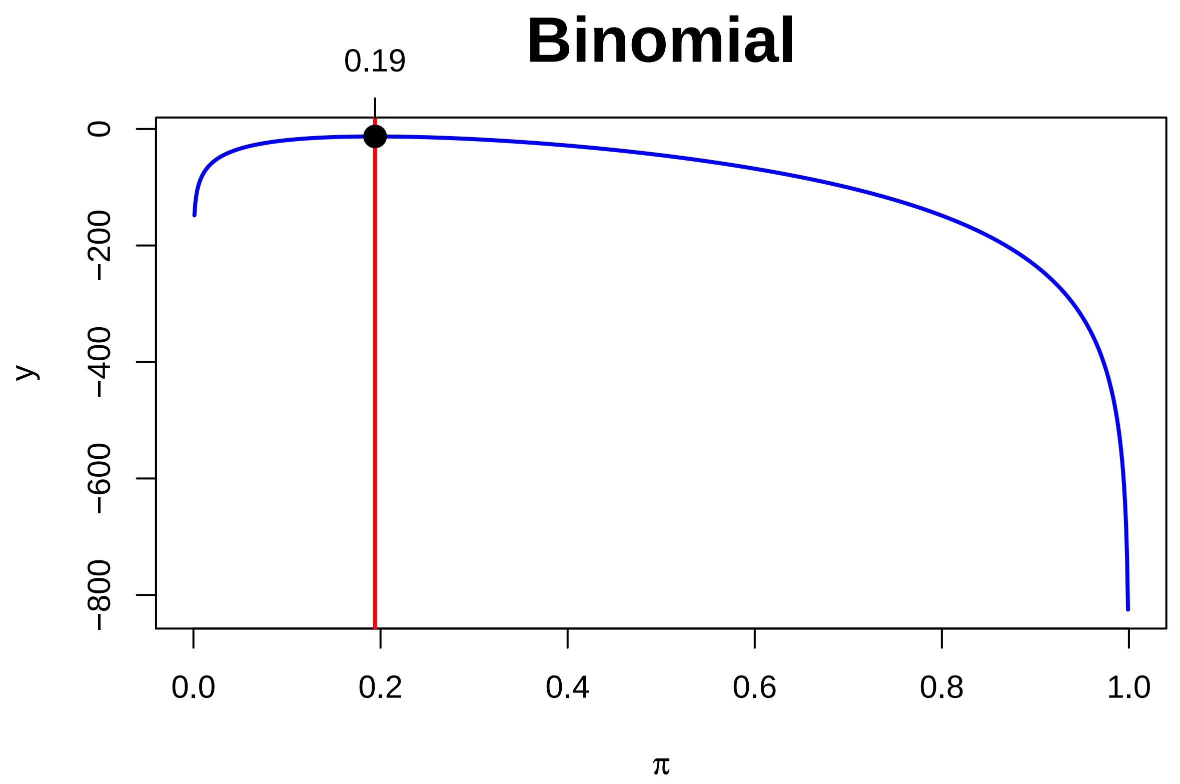
```r
getwd()
```

```
## [1] "C:/Users/cglen/Desktop/Stat Methods/Labs/LAB10"
```

Task2

$$L(p;x) = \frac{20!}{x!(20-x)!}p^x(1-p)^{20-x}$$

```r
logbin=function(x,param) log(dbinom(x,prob=param,size=20))
mymaxlik=function(lfun,x,param,...){
  np=length(param)
  z=outer(x,param,lfun)
  y=apply(z,2,sum)
  plot(param,y,col="Blue",type="l",lwd=2,...)
  i=max(which(y==max(y)))
  abline(v=param[i],lwd=2,col="Red")
  points(param[i],y[i],pch=19,cex=1.5,col="Black")
  axis(3,param[i],round(param[i],2))
  ifelse(i-3>=1 & i+2<=np, slope<-(y[(i-2):(i+2)]-y[(i-3):(i+1)])/(param[(i-2):(i+2)]-param[(i-3):(i+1)]
  return(list(i=i,parami=param[i],yi=y[i],slope=slope))
}
mymaxlik(x=c(3,3,4,3,4,5,5,4),param=seq(0,1,length=1000),lfun=logbin,xlab=expression(pi),main="Binomial
```



```
## $i
## [1] 195
##
## $parami
## [1] 0.1941942
##
## $yi
## [1] -12.78734
##
## $slope
## [1]  2.12579310  1.08781082  0.05802047 -0.96371372 -1.97752478
```

Task3

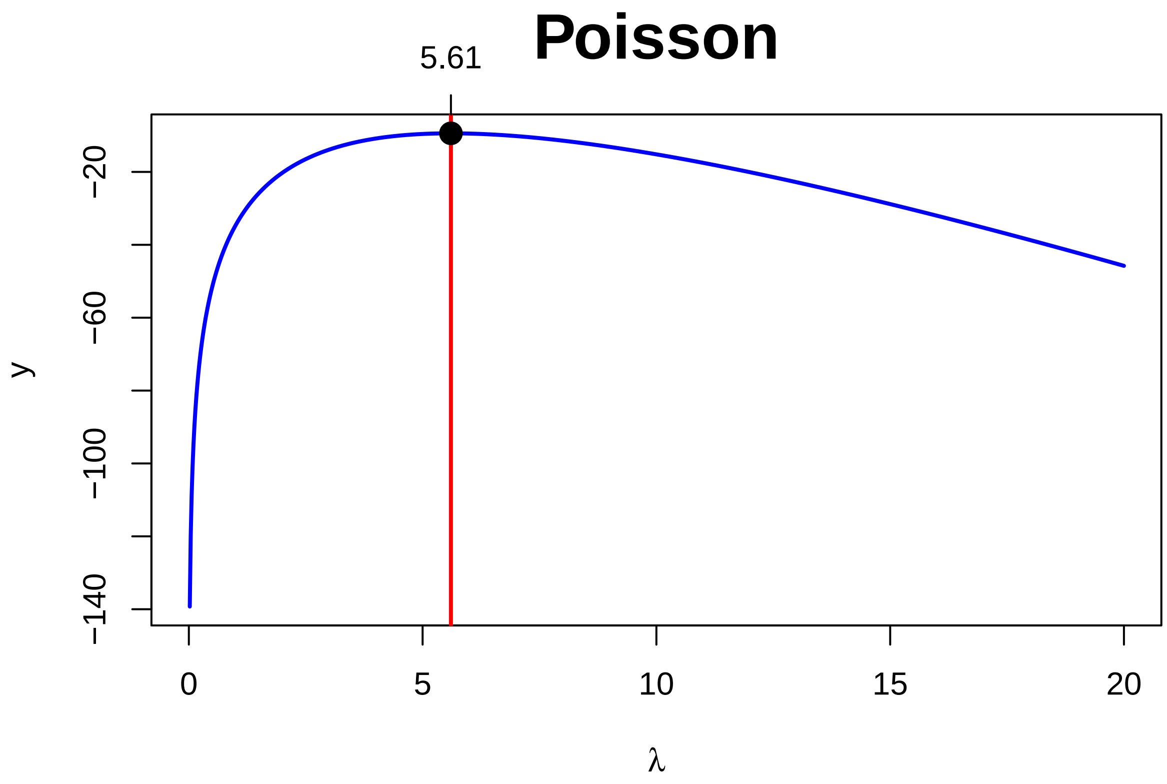
$$P(y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

$$L(\lambda : y) = \frac{e^{-n\lambda}\lambda^{\sum_{t}^{n}=0 y_t}}{x_1!x_2!\cdots x_n!}$$

$$l(\lambda : y) = log(L(\lambda : y)) = log(e^{-n\lambda}) + log(\lambda^{\sum_{t=0}^{n} y_t})$$

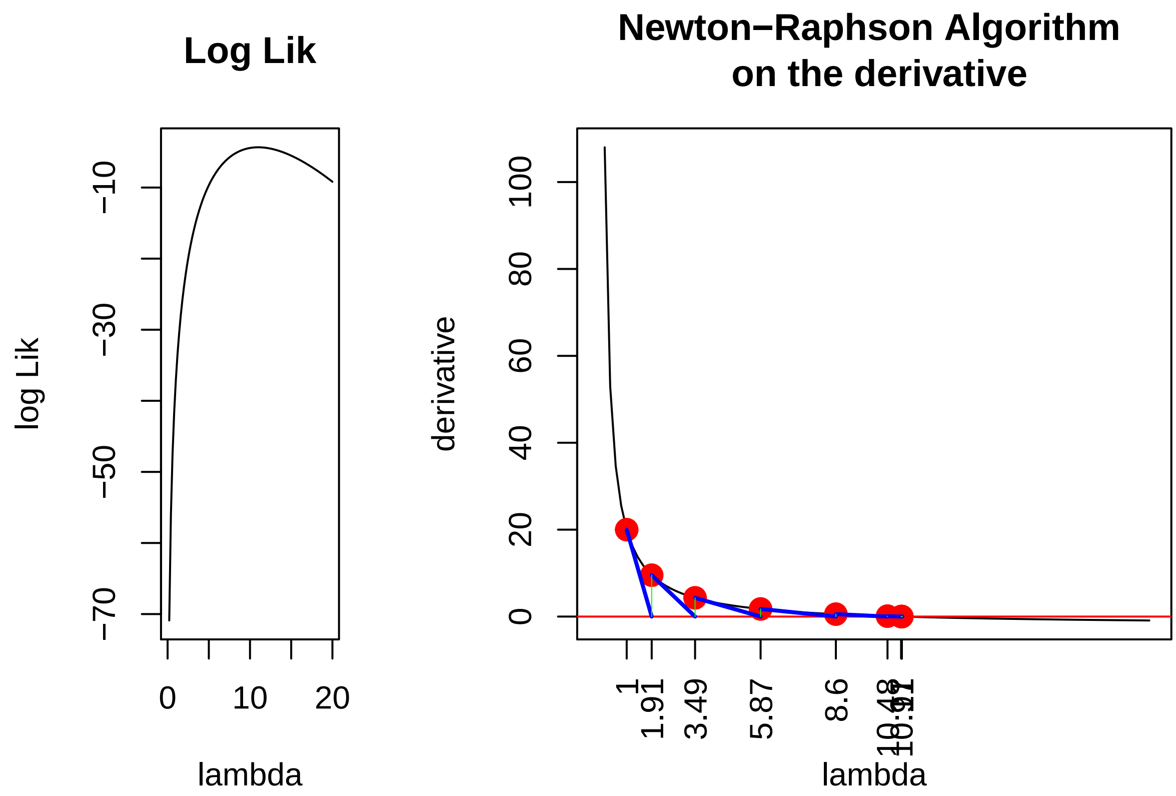
$$l(\lambda : y) = -n\lambda log(e) + \sum_{t=0}^{n} y_t log(\lambda) = \sum_{t=0}^{n} y_t log(\lambda) - n\lambda$$

```
logpoiss=function(x,param) log(dpois(x,lambda=param))
mymaxlik(x=c(4,6,7,6,5),param=seq(0,20,length=1000),lfun=logpoiss,xlab=expression(lambda),main="Poisson
```



```
## $i
## [1] 281
##
## $parami
## [1] 5.605606
##
## $yi
## [1] -9.411759
##
## $slope
## [1]  0.040005454  0.021908448  0.003940937 -0.013898462 -0.031611116
```

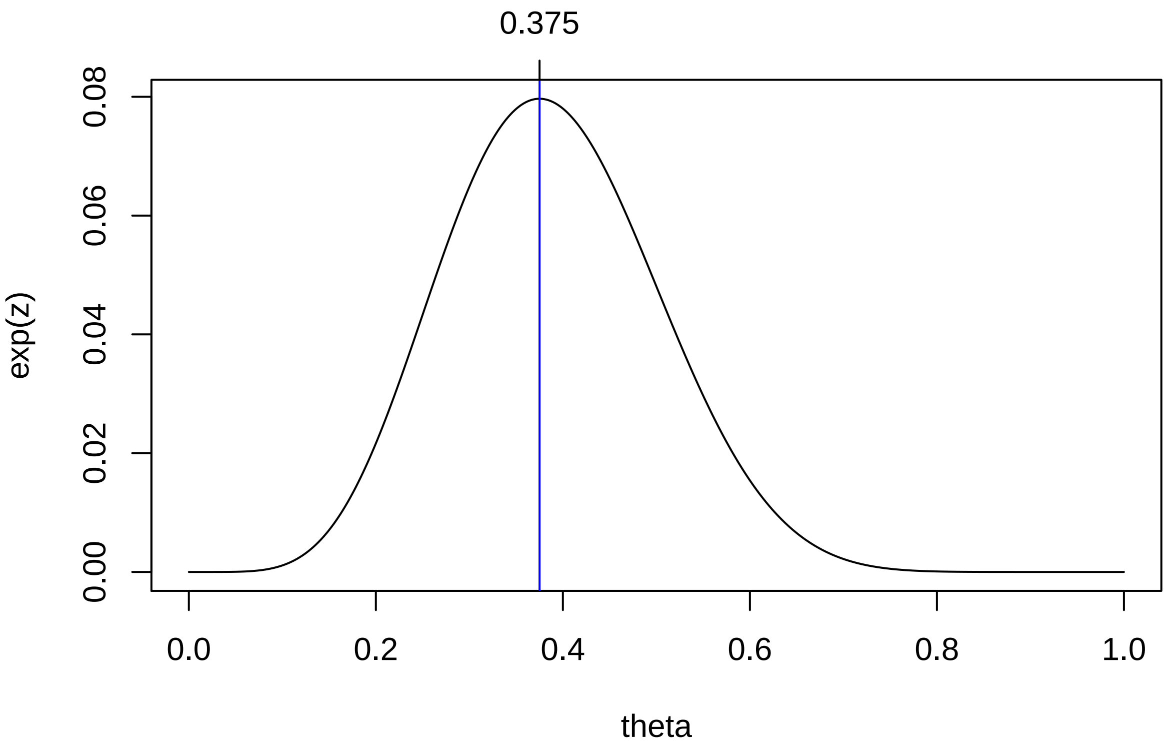
```
myNRML=function(x0,delta=0.001,llik,xrange,parameter="param"){
  f=function(x) (llik(x+delta)-llik(x))/delta
  fdash=function(x) (f(x+delta)-f(x))/delta
  d=1000
  i=0
  x=c()
  y=c()
  x[1]=x0
  y[1]=f(x[1])
  while(d > delta & i<100){
    i=i+1
    x[i+1]=x[i]-f(x[i])/fdash(x[i])
    y[i+1]=f(x[i+1])
    d=abs(x[i+1]-x[i])
  }
  layout(matrix(1:2,nr=1,nc=2,byrow=TRUE),width=c(1,2))
  curve(llik(x), xlim=xrange,xlab=parameter,ylab="log Lik",main="Log Lik")
  curve(f(x),xlim=xrange,xaxt="n", xlab=parameter,ylab="derivative",main=  "Newton-Raphson Algorithm \n
  points(x,y,col="Red",pch=19,cex=1.5)
  axis(1,x,round(x,2),las=2)
  abline(h=0,col="Red")
  segments(x[1:(i-1)],y[1:(i-1)],x[2:i],rep(0,i-1),col="Blue",lwd=2)
  segments(x[2:i],rep(0,i-1),x[2:i],y[2:i],lwd=0.5,col="Green")
  list(x=x,y=y)
}
myNRML(x0=1,delta=0.000001,llik=function(x) log(dpois(12,x)*dpois(10,x)),xrange=c(0,20),parameter="lamb
```



```
## $x
##  [1]  1.000000  1.908863  3.485981  5.867603  8.603229 10.481457 10.974386
##  [8] 11.000023 11.000000 10.999999
##
## $y
##  [1]  1.999999e+01  9.525183e+00  4.310992e+00  1.749401e+00  5.571793e-01
##  [6]  9.894478e-02  4.667942e-03 -4.244605e-06 -6.217249e-09  8.881784e-10
```

Task4

```
logbin2=function(theta){log(dbinom(2,prob=theta,size=6)) + log(dbinom(4,prob=theta,size=10))}
mymaxlikg=function(lfun="logbin2",theta) {
  nth=length(theta)
  thmat=matrix(theta,nr=nth,nc=1,byrow=TRUE)
  z=apply(thmat,1,lfun)
  zmax=max(which(z==max(z)))
  plot(theta,exp(z),type="l")
  abline(v=theta[zmax],col="Blue")
  axis(3,theta[zmax],round(theta[zmax],4))
  theta[zmax]
}
mymaxlikg(theta=seq(0,1,length=10000))
```



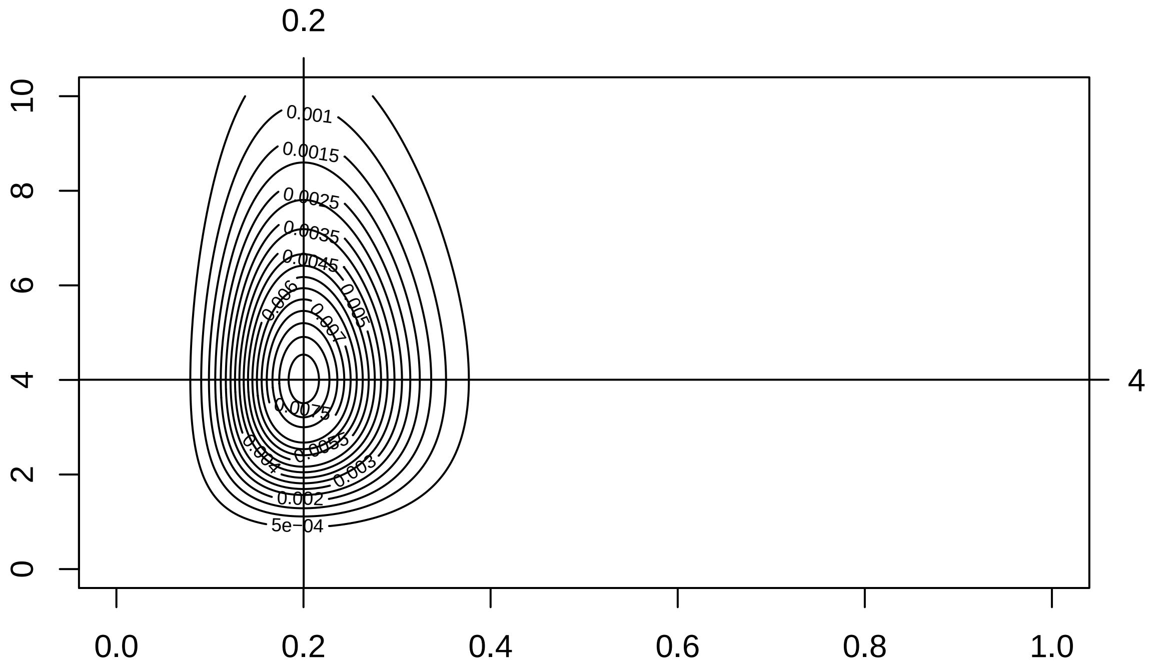
```
## [1] 0.3750375
```

Task5

$$l(\theta_1|\theta_2) = \sum_{t=0}^{n} y_t log(\lambda) - n\lambda - 2log(x!) + log(n!) - log((n-x)!) + xlog(p) + (n-k)log(1-p)$$

```
logbinpois=function(theta1,theta2) log(dbinom(4,size=20,prob=theta1)) + log(dbinom(4,size=20,prob=theta
```

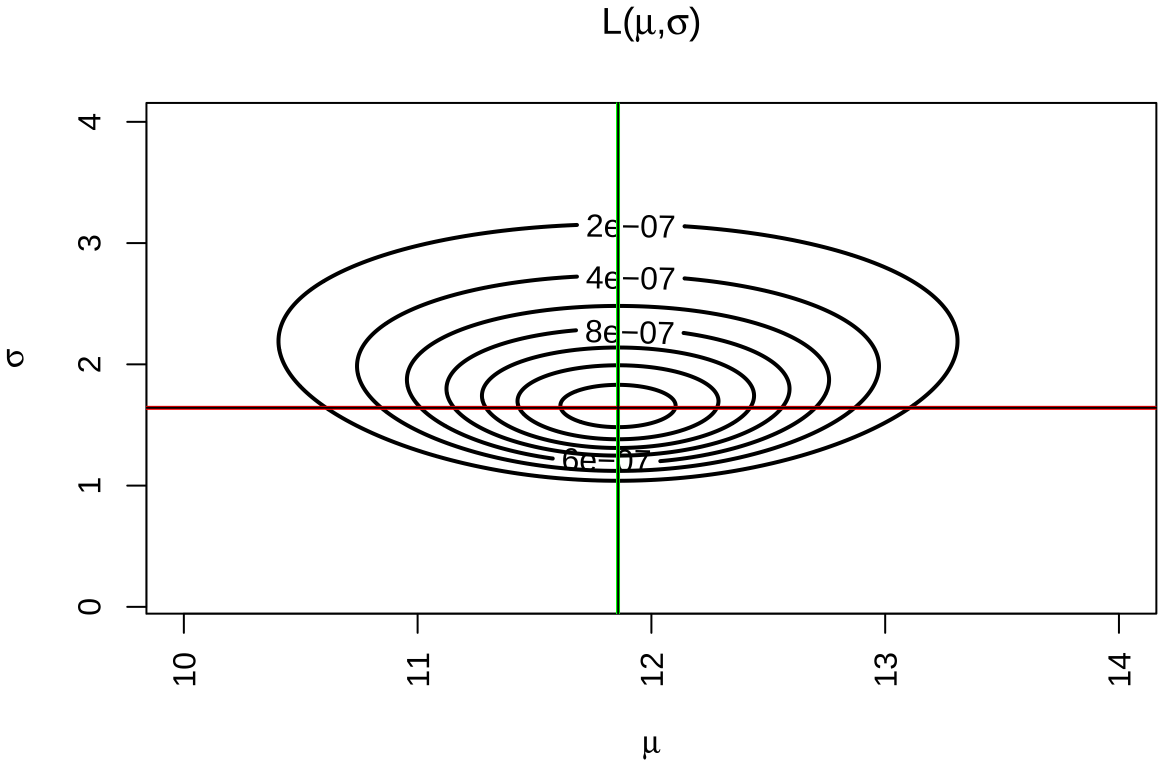
```r
maxlikg2=function(theta1,theta2,lfun="logbinpois",...){
  n1=length(theta1)
  n2=length(theta2)
  z=outer(theta1,theta2,lfun)
  contour(theta1,theta2,exp(z),...)
  maxl=max(exp(z))
  coord=which(exp(z)==maxl,arr.ind=TRUE)
  th1est=theta1[coord[1]]
  th2est=theta2[coord[2]]
  abline(v=th1est,h=th2est)
  axis(3,th1est,round(th1est,2))
  axis(4,th2est,round(th2est,2),las=1)
  list(th1est=th1est,th2est=th2est)
}
maxlikg2(theta1=seq(0,1,length=1000),theta2=seq(0,10,length=1000),nlevels=20)
```



```
## $th1est
## [1] 0.2002002
##
## $th2est
## [1] 4.004004
```

Task6

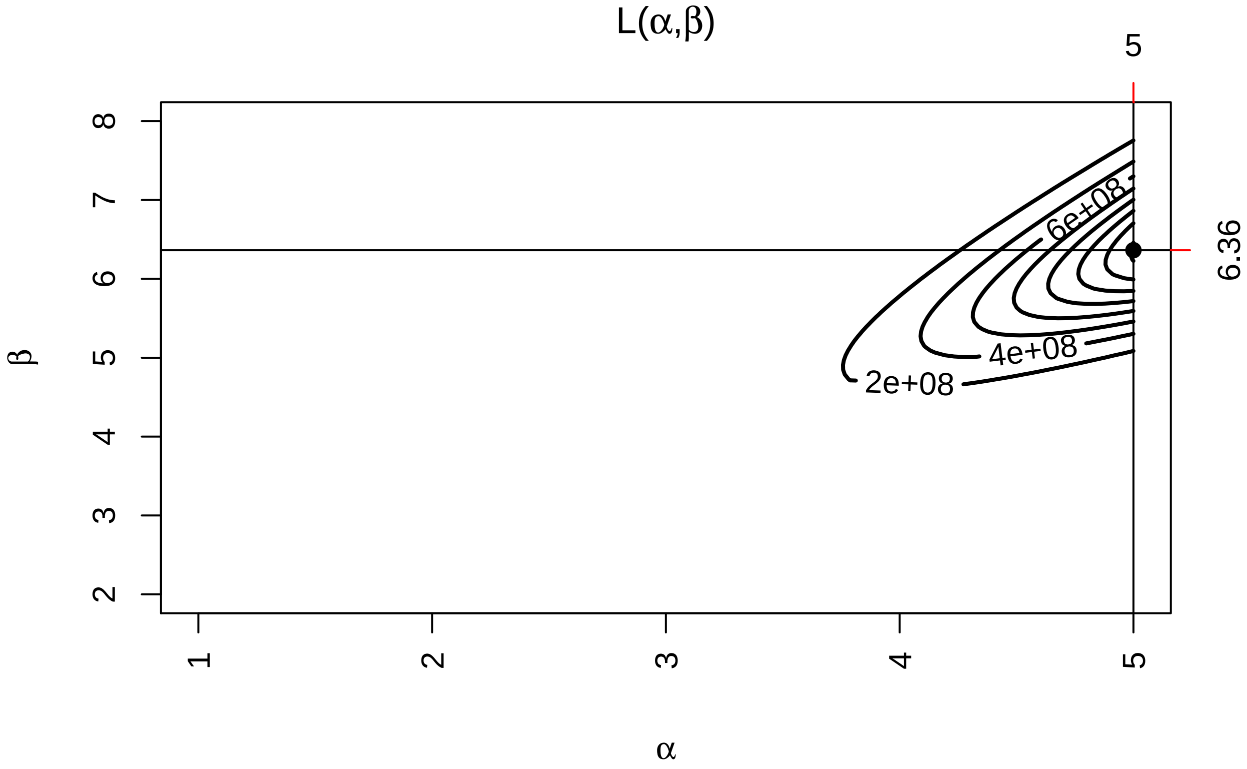
```r
mymlnorm=function(x,mu,sig,...){
  nmu=length(mu)
  nsig=length(sig)
  n=length(x)
  zz=c()
  lfun=function(x,m,p) log(dnorm(x,mean=m,sd=p))
  for(j in 1:nsig){
    z=outer(x,mu,lfun,p=sig[j])
    y=apply(z,2,sum)
    zz=cbind(zz,y)
  }
  maxl=max(exp(zz))
  coord=which(exp(zz)==maxl,arr.ind=TRUE)
  maxlsig=apply(zz,1,max)
  contour(mu,sig,exp(zz),las=3,xlab=expression(mu),ylab=expression(sigma),axes=TRUE,
          main=expression(paste("L(",mu,",",sigma,")",sep="")),...)
  mlx=round(mean(x),2)
  mly=round(sqrt((n-1)/n)*sd(x),2)
  abline(v=mean(x),lwd=2,col="Green")
  abline(h=sqrt((n-1)/n)*sd(x),lwd=2,col="Red")
  muest=mu[coord[1]]
  sigest=sig[coord[2]]
  abline(v=muest, h=sigest)
  return(list(x=x,coord=coord,maxl=maxl))
}

mymlnorm(x=c(10,12,13,15,12,11,10),mu=seq(10,14,length=1000),sig=seq(0.1,4,length=1000),lwd=2,labcex=1)
```



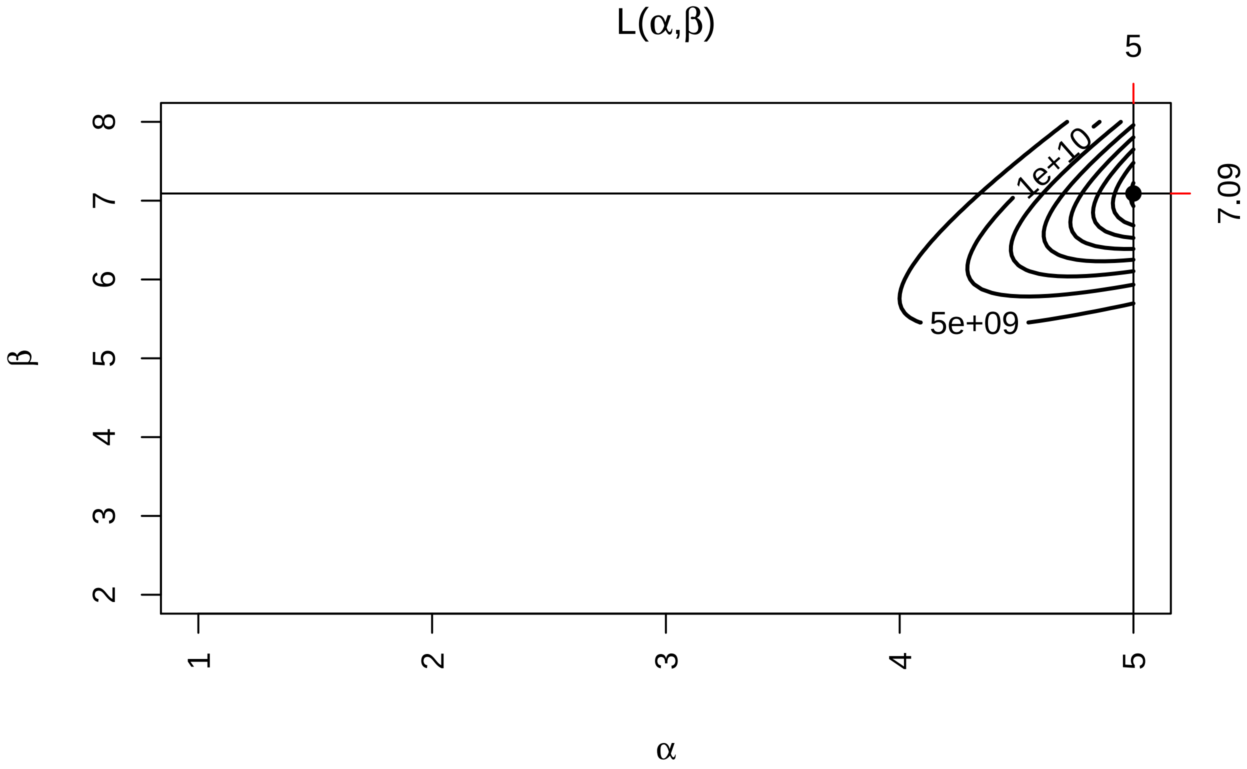
```
## $x
## [1] 10 12 13 15 12 11 10
##
## $coord
##      row col
## [1,] 465 396
##
## $maxl
## [1] 1.513618e-06
```

Task7

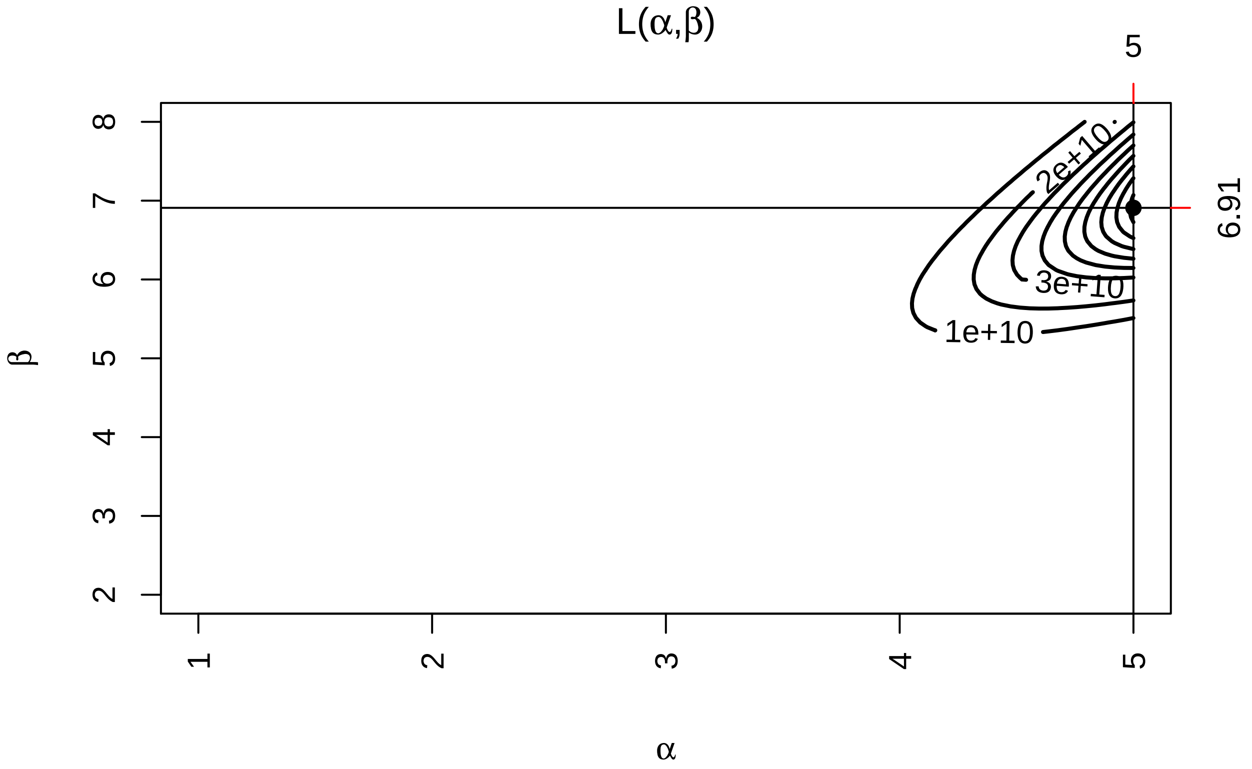
```
mymlbeta=function(x,alpha,beta,...){
  na=length(alpha)
  nb=length(beta)
  n=length(x)
  zz=c()
  lfun=function(x,a,b) log(dbeta(x,shape1=a,shape2=b))
  for(j in 1:nb){
    z=outer(x,alpha,lfun,b=beta[j])
    y=apply(z,2,sum)
    zz=cbind(zz,y)
  }
  maxl=max(exp(zz))
  coord=which(exp(zz)==maxl,arr.ind=TRUE)
  aest=alpha[coord[1]]
  best=beta[coord[2]]
  contour(alpha,beta,exp(zz),las=3,xlab=expression(alpha),ylab=expression(beta),axes=TRUE,
          main=expression(paste("L(",alpha,",",beta,")",sep="")),...)
  abline(v=aest, h=best)
  points(aest,best,pch=19)
  axis(4,best,round(best,2),col="Red")
  axis(3,aest,round(aest,2),col="Red")
  return(list(x=x,coord=coord,maxl=maxl,maxalpha=aest,maxbeta=best))
}
x <- mymlbeta(x=rbeta(30,shape1=3,shape2=4),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,la
```



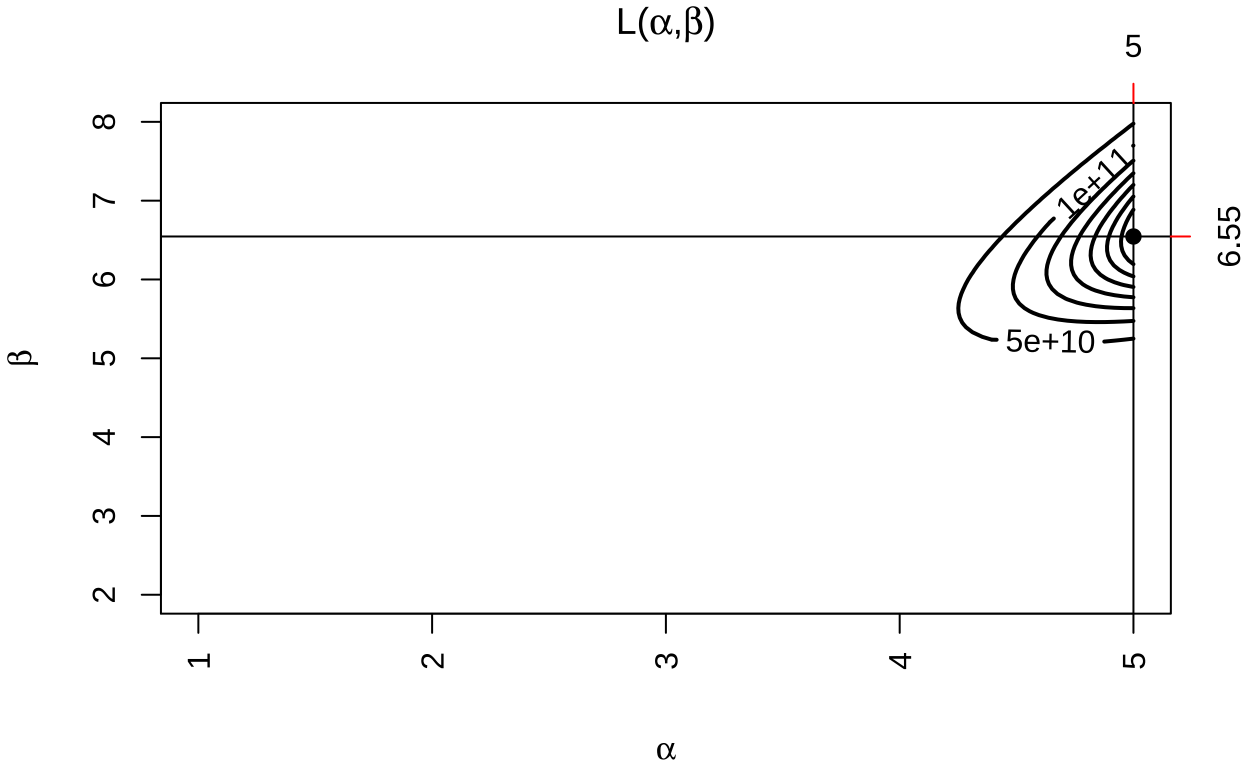
```
ax<-x$maxalpha
ay <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



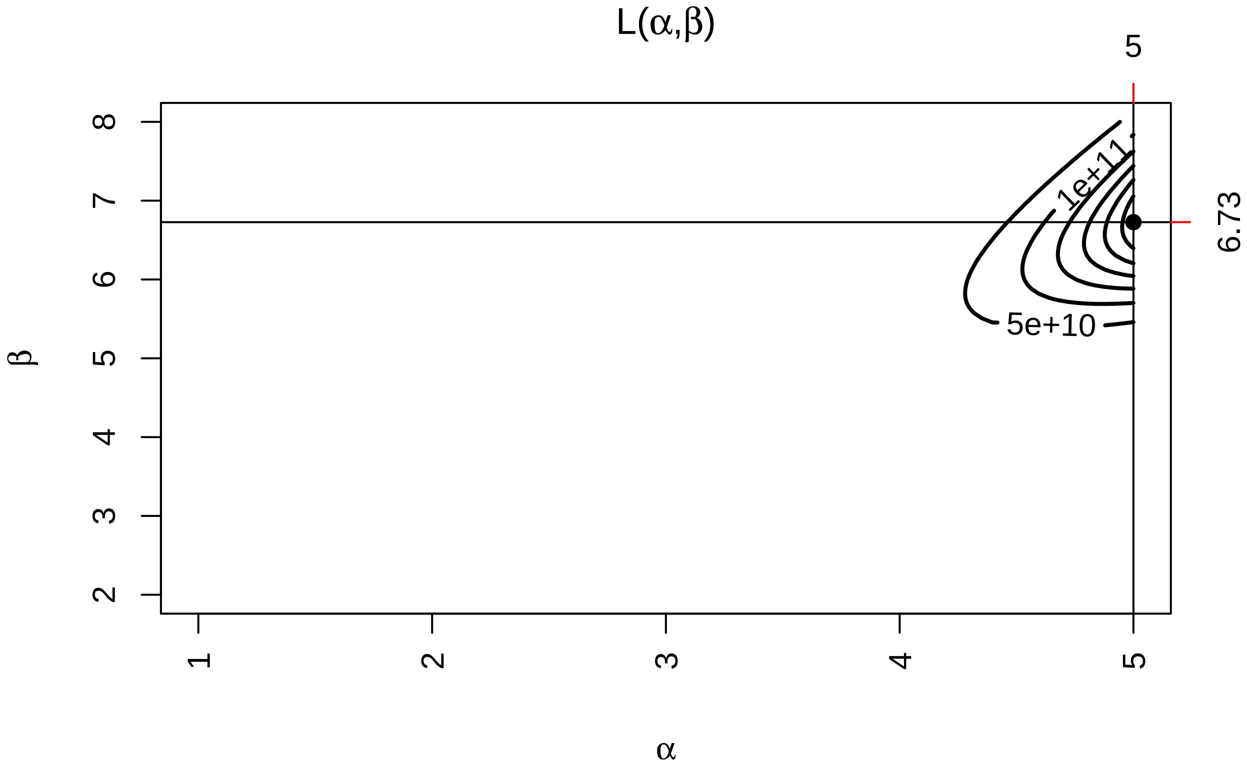
```
bx<-x$maxalpha
by <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



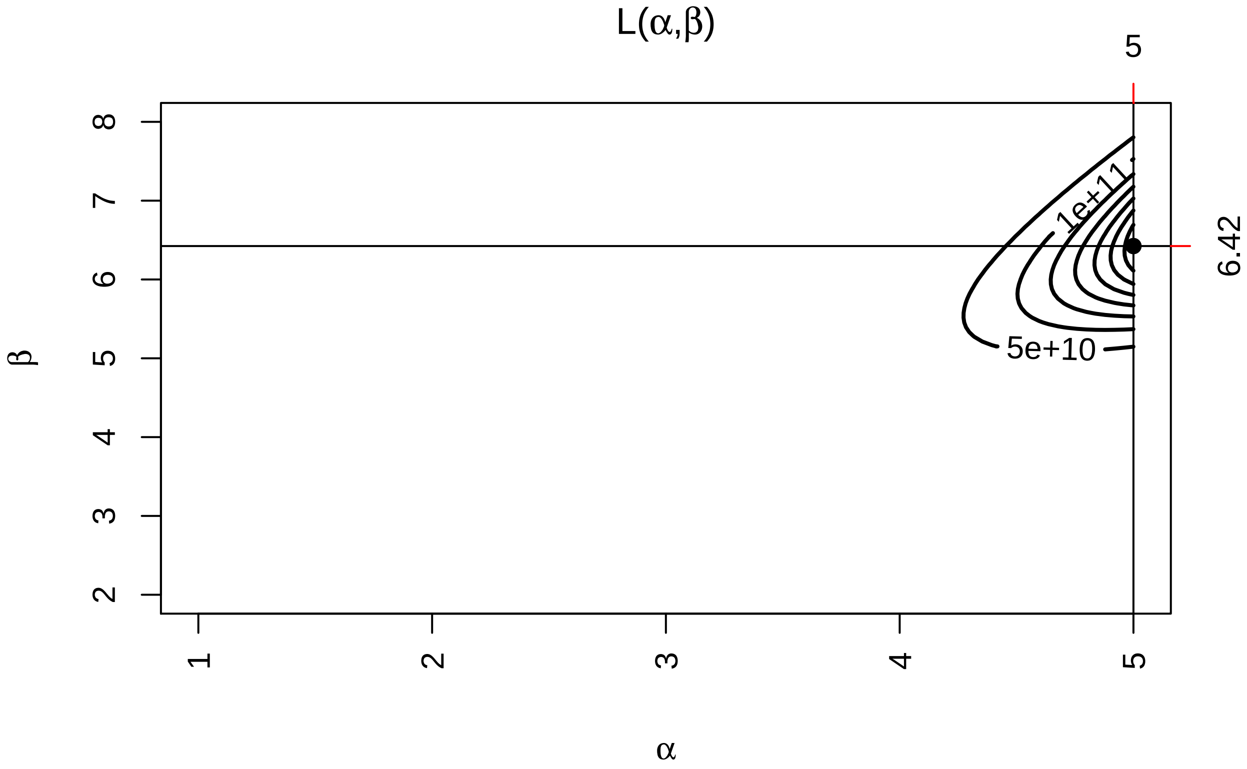
```
cx<-x$maxalpha
cy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



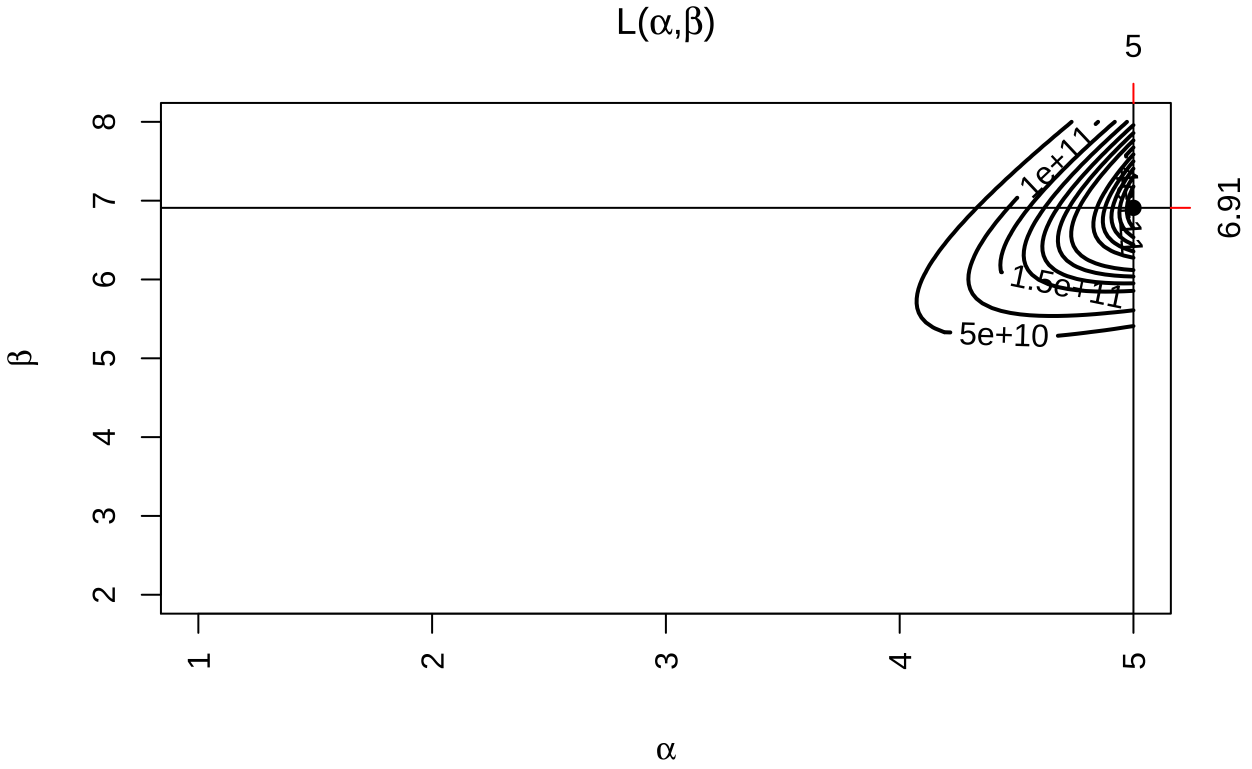
```
dx<-x$maxalpha
dy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



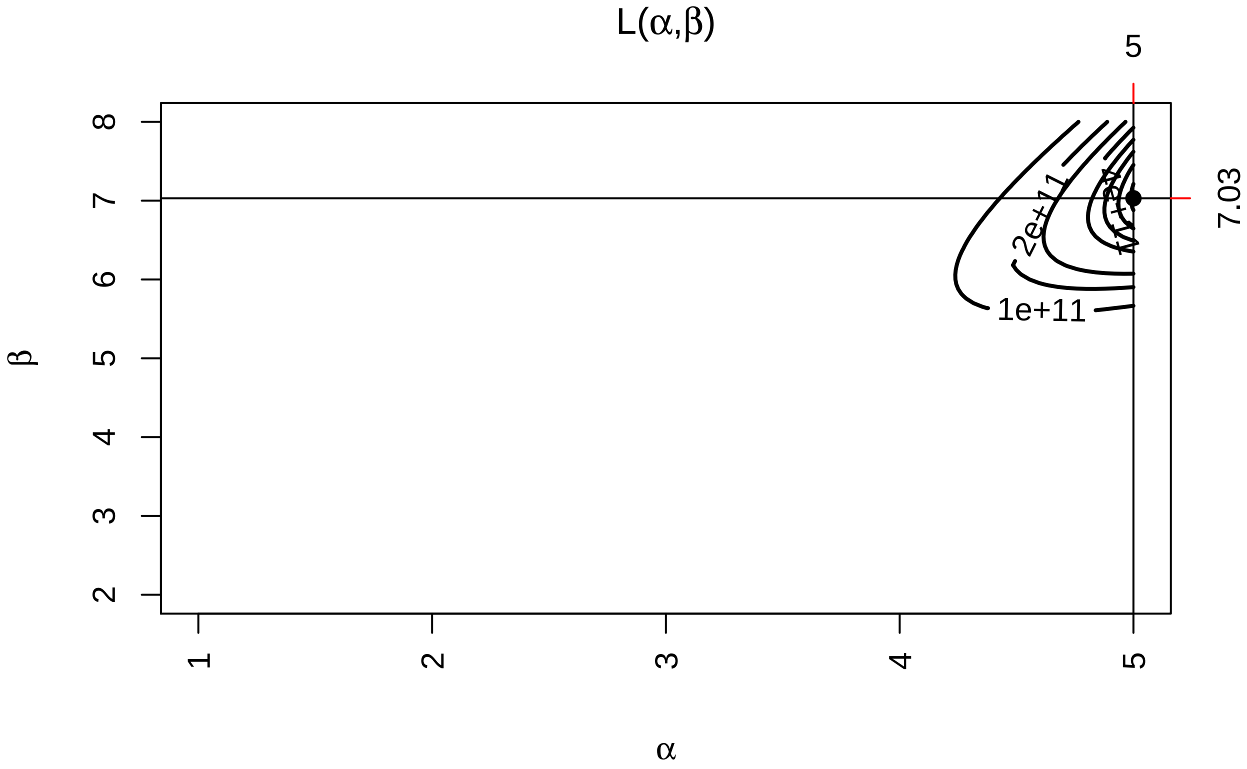
```
ex<-x$maxalpha
ey <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



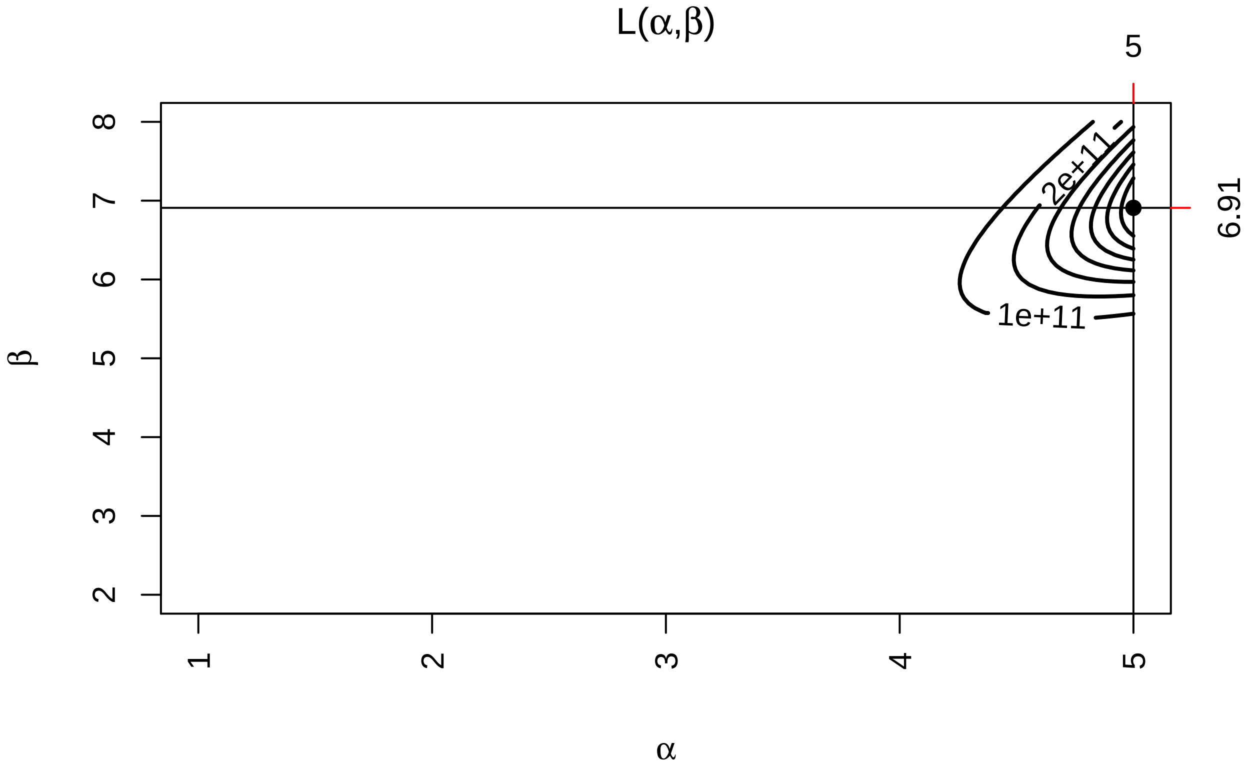
```
fx<-x$maxalpha
fy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



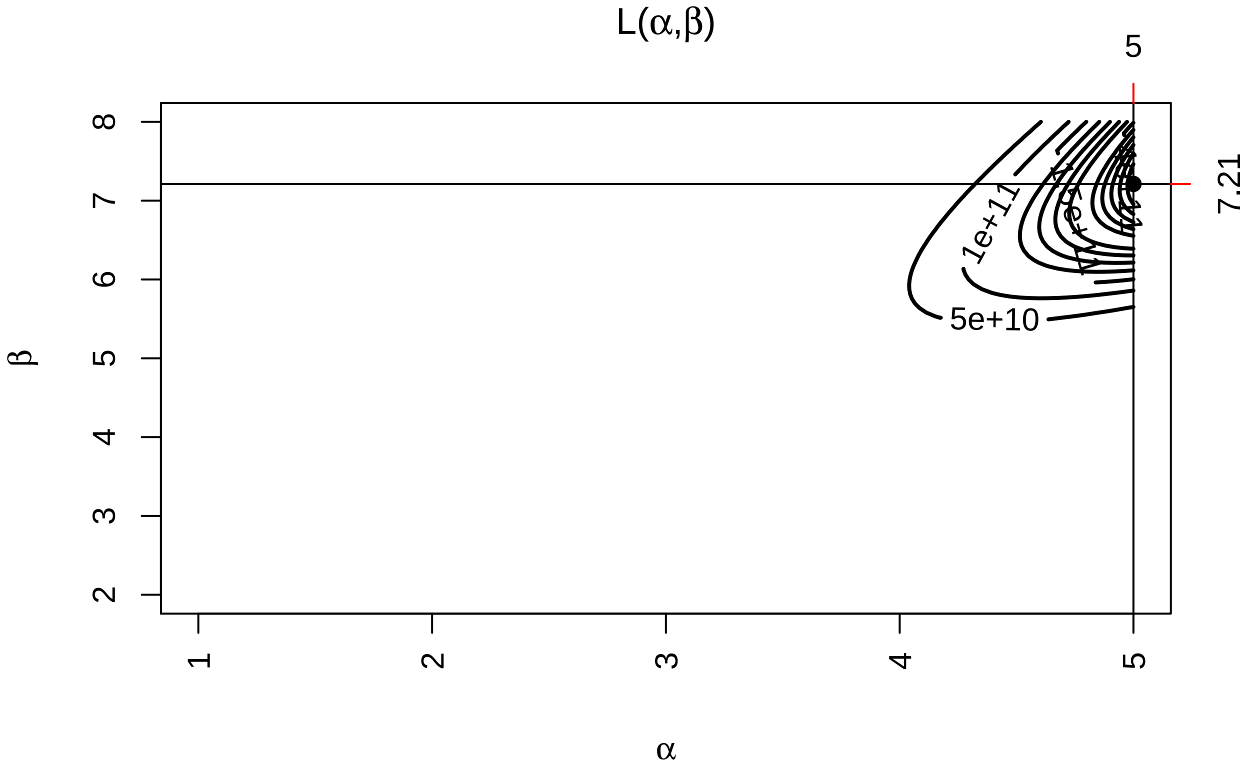
```
gx<-x$maxalpha
gy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



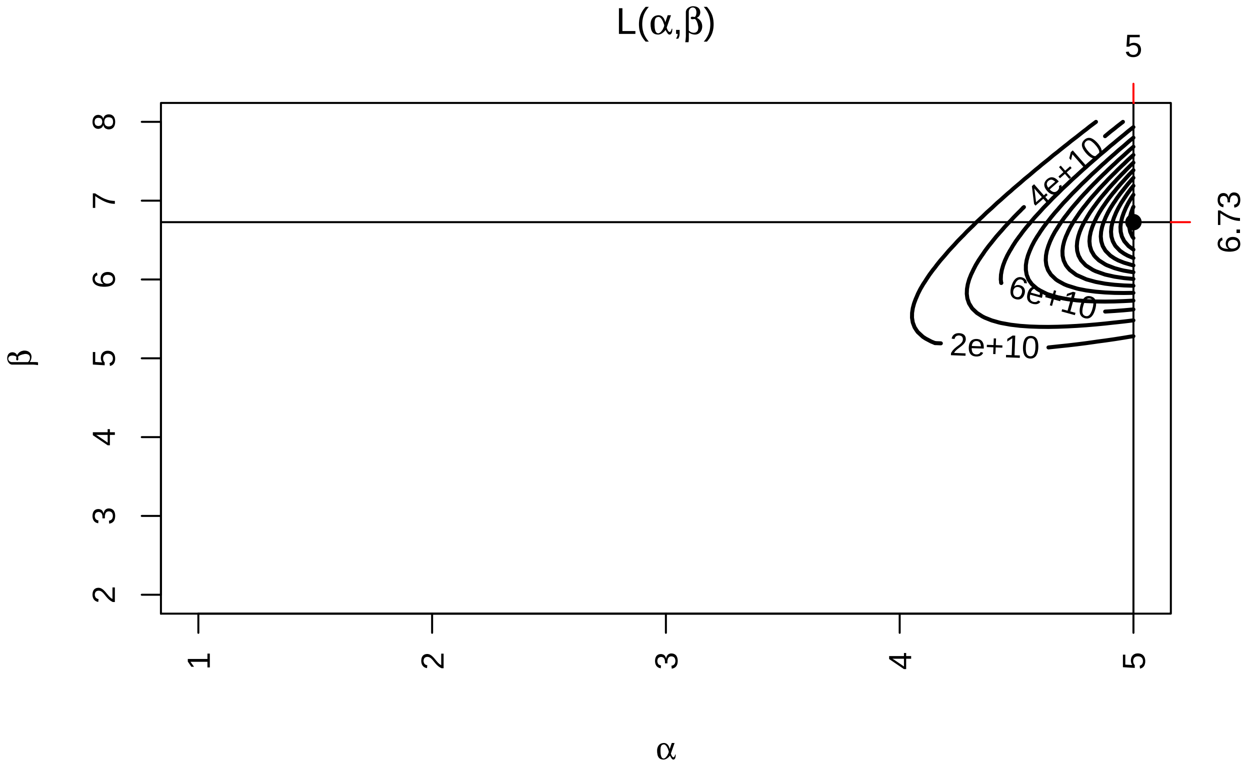
```
hx<-x$maxalpha
hy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



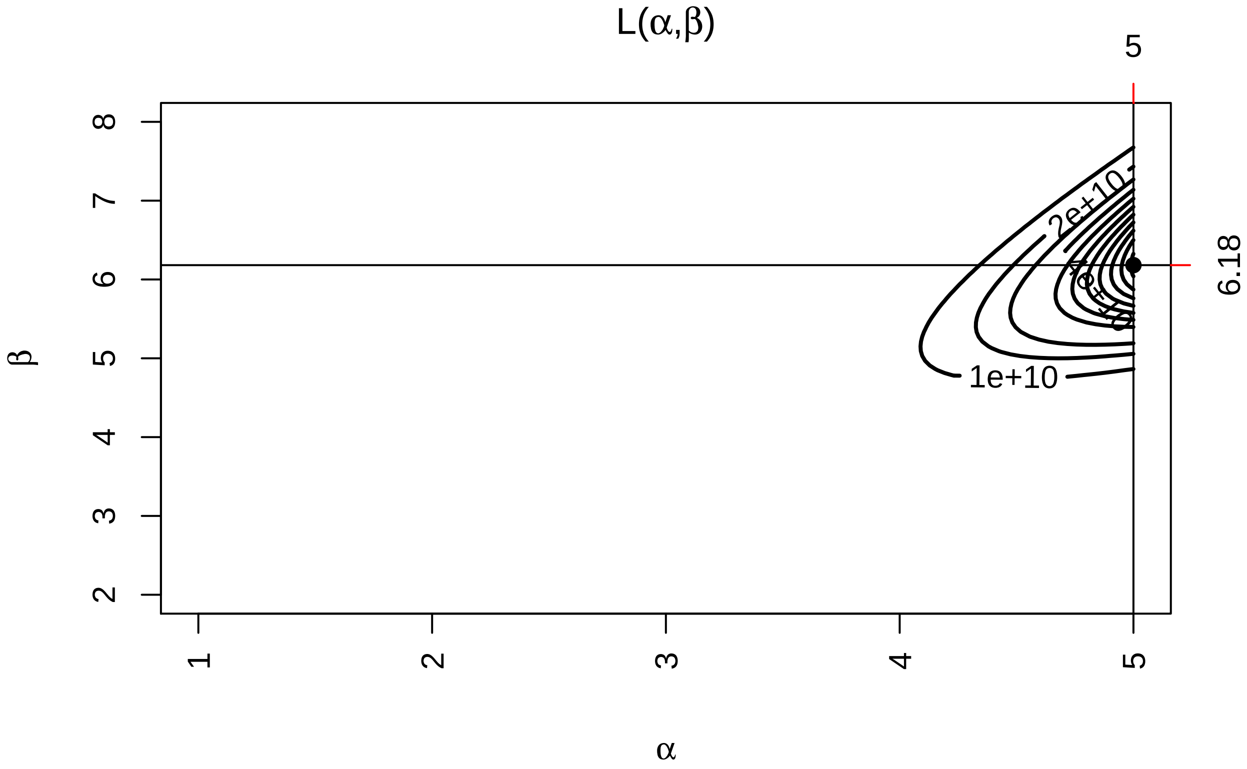
```
ix<-x$maxalpha
iy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



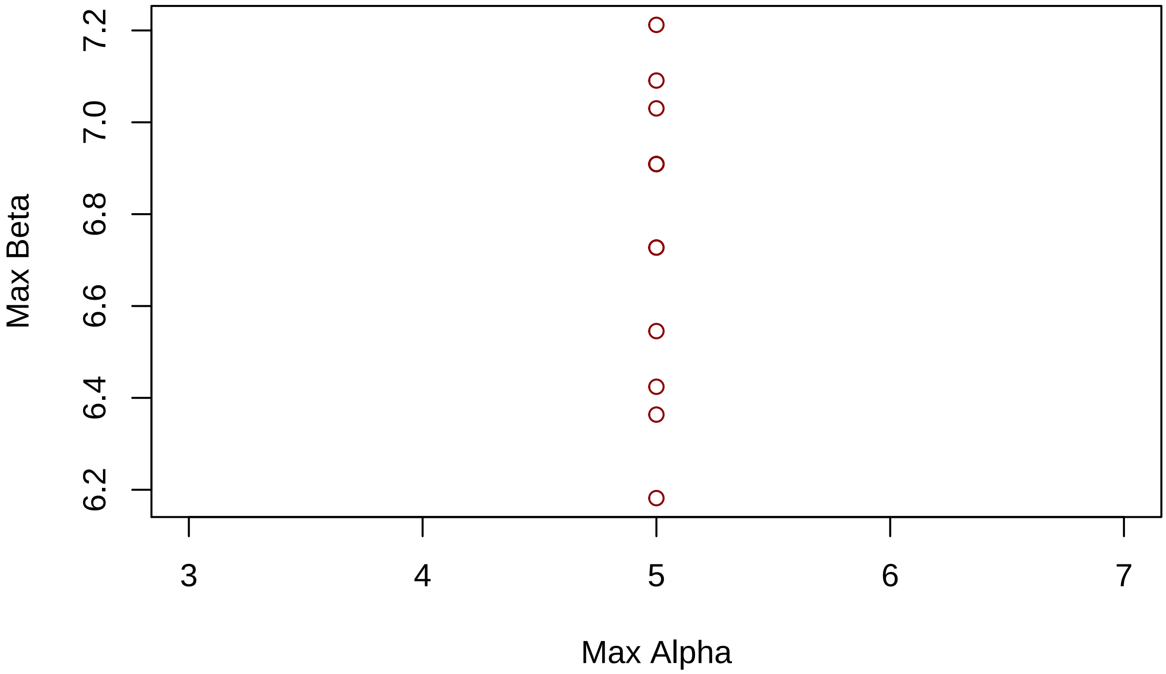
```
jx<-x$maxalpha
jy <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



```
kx<-x$maxalpha
ky <- x$maxbeta
x <- mymlbeta(x=sample(x$x, replace=TRUE),alpha=seq(1,5,length=100),beta=seq(2,8,length=100),lwd=2,labc
```



```
lx<-x$maxalpha
ly <- x$maxbeta
plot(c(ay,by,cy,dy,ey,fy,gy,hy,iy,jy,ky,ly)~c(ax,bx,cx,dx,ex,fx,gx,hx,ix,jx,kx,lx), ylab = "Max Beta", 
```


Max Beta
7.2
7.0
6.8
6.6
6.4
6.2
3
4
5
6
7
Max Alpha