

## R作业4程序及结果(仅供参考)

April 5, 2016

1.

```
m=10000
x1=rnorm(m);x2=rnorm(m)
mean(abs(x1-x2))

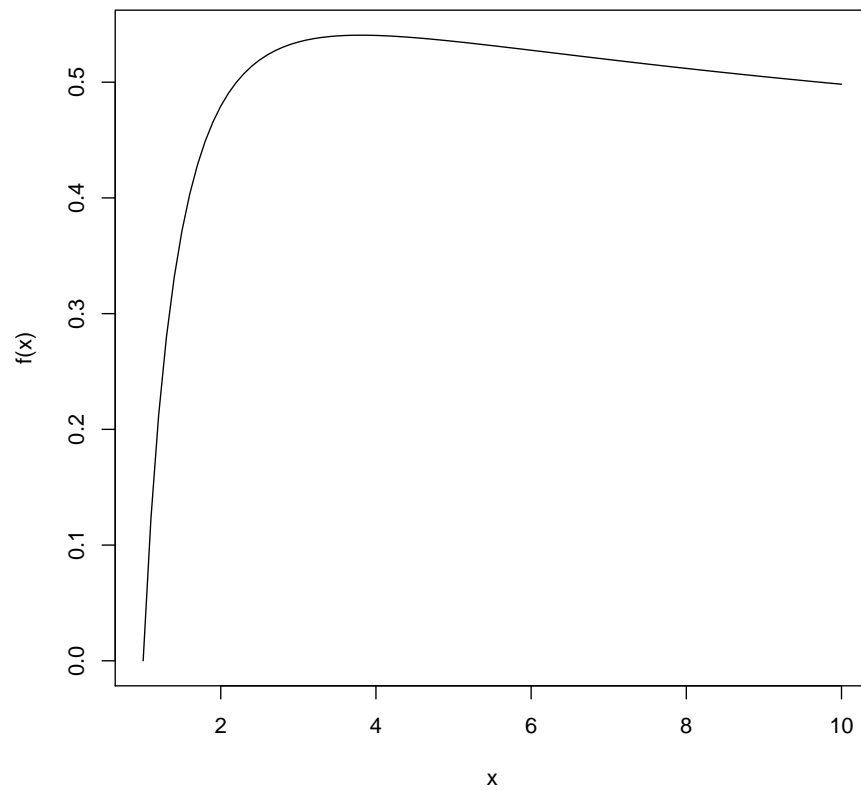
[1] 1.134709

var(abs(x1-x2))

[1] 0.7317431
```

2.

```
f=function(x) log(1+log(x))/log(1+x)
x=seq(1,10,by=0.1)
plot(x,f(x),type="l")
```



```
optimize(f,c(1,10),maximum=T) #求最大值
```

```
$maximum  
[1] 3.803908
```

```
$objective  
[1] 0.5406119
```

3.  
(1)

```
set.seed(100)  
x=rgamma(100,4,2)
```

(2)

```

n=length(x)
A1=mean(x);M2=(n-1)/n*var(x)
beta.m=A1/M2;beta.m

[1] 2.350384

alpha.m=beta.m*A1;alpha.m

[1] 4.548178

```

(3)  
方法1:nlm求-对数似然函数的最小值

```

nlogl=function(pr){
  loglik=n*pr[1]*log(pr[2])-n*log(gamma(pr[1]))+(pr[1]-1)*sum(log(x))-pr[2]*sum(x)
  return(-loglik)
}
pr0=c(1,2)
nlm(nlogl,pr0)

Warning in log(pr[2]): 产生了NaNs
Warning in f(x, ...): 'gammafn'里的值在范围外
Warning in nlm(nlogl, pr0): NA/Inf被换成最大的正值
Warning in log(pr[2]): 产生了NaNs
Warning in nlm(nlogl, pr0): NA/Inf被换成最大的正值

$minimum
[1] 127.1864

$estimate
[1] 4.256207 2.199501

$gradient
[1] 2.773653e-06 -5.216695e-06

$code
[1] 1

$iterations
[1] 14

```

方法2:用uniroot解对数似然方程

```

f=function(alpha){
  n*log(alpha/mean(x))-n*digamma(alpha)+sum(log(x))
}
uniroot(f,c(1,10))

```

```

$root
[1] 4.256183

$f.root
[1] 7.241747e-05

$iter
[1] 7

$init.it
[1] NA

$estim.prec
[1] 6.103516e-05

```

4.  
(1)

```

intval=function(x,alpha=0.05){
  n=length(x);df=n-1
  a=(n-1)*var(x)/qchisq(1-alpha/2,df)
  b=(n-1)*var(x)/qchisq(alpha/2,df)
  data.frame(a=a,b=b)
}

```

(2)

```

n=100
x=rnorm(n,2,2)
intval(x)

      a      b
1 2.959031 5.179922

```

(3)

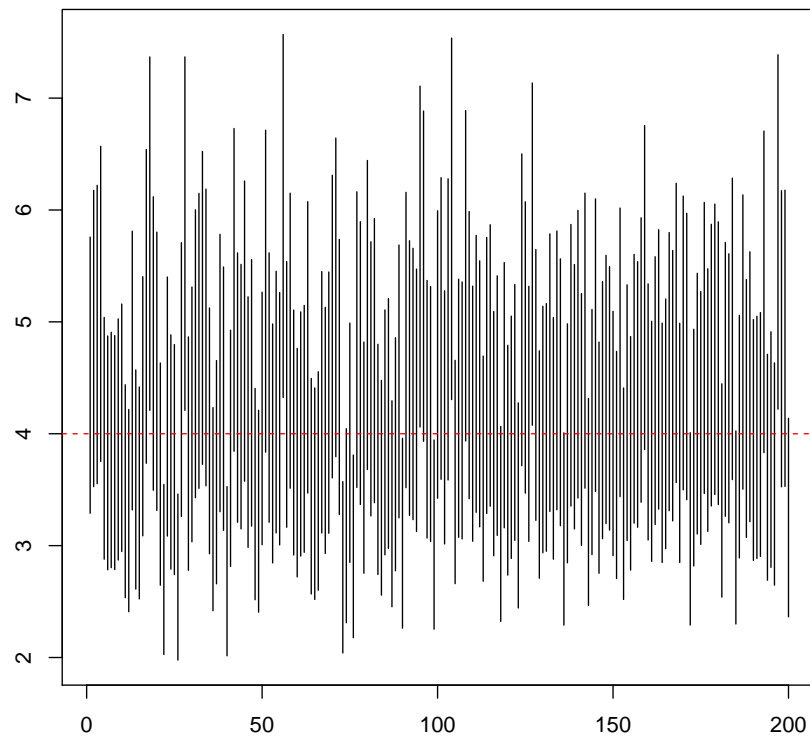
```

m=200
left=numeric(m);right=numeric(m)
for(i in 1:m){
  x=rnorm(n,2,2)
  estimate=intval(x)
  left[i]=estimate$a
  right[i]=estimate$b
}
sum(left<4&right>4)

```

```
[1] 186
```

```
min=min(left)
max=max(right)
plot(1:m,right,type="n",ylim=c(min,max),xlab="",ylab="")
segments(1:m,left,1:m,right)
abline(h=4,lty=2,col=2)
```



5. (1)

```
intval.mu=function(x,sigma=-1,alpha=0.05){
  xb=mean(x)
  if(sigma>0) tmp=sigma/sqrt(n)*qnorm(1-alpha/2)
  else tmp=sd(x)/sqrt(n)*qt(1-alpha/2,n-1)
  left=xb-tmp;right=xb+tmp;len=right-left
  data.frame(left=left,right=right,len=len)
}
```

(2)

```
n=100
x=rnorm(n,2,2)
int1=intval.mu(x);int1

      left    right    len
1 1.688548 2.496625 0.808077

int2=intval.mu(x,sigma=2);int2

      left    right    len
1 1.700593 2.484579 0.7839856

int1$len>int2$len

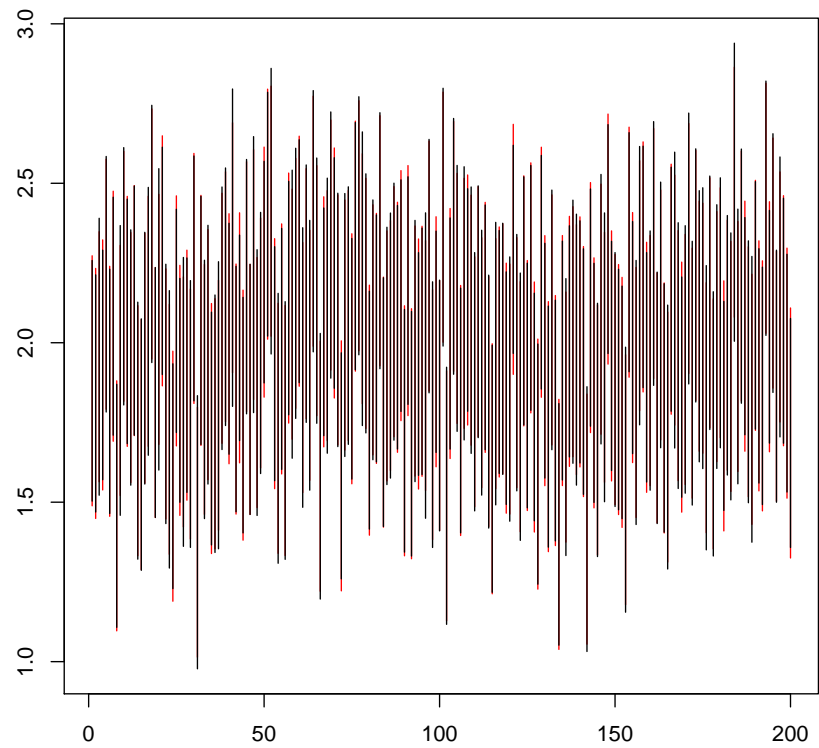
[1] TRUE
```

(3)

```
m=200
interval.known=matrix(0,m,3)
interval.unknown=matrix(0,m,3)
for(i in 1:m){
  x=rnorm(n,2,2)
  int1=intval.mu(x)
  interval.unknown[i,1]=int1$left
  interval.unknown[i,2]=int1$right
  interval.unknown[i,3]=int1$len

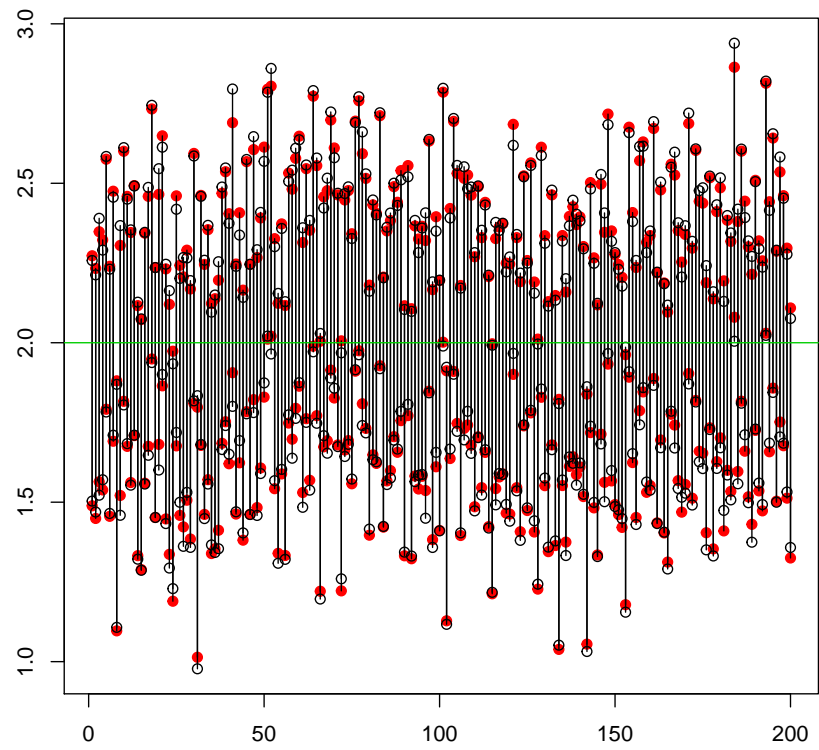
  int2=intval.mu(x,sigma=2)
  interval.known[i,1]=int2$left
  interval.known[i,2]=int2$right
  interval.known[i,3]=int2$len
}

min=min(interval.known[,1],interval.unknown[,1])
max=max(interval.known[,2],interval.unknown[,2])
plot(1:m,interval.known[,1],ylim=c(min,max),type="n",xlab="",ylab="")
segments(1:m,interval.known[,1],1:m,interval.known[,2],col=2)
segments(1:m,interval.unknown[,1],1:m,interval.unknown[,2])
```



Another plot

```
plot(1:m,interval.known[,1],pch=19,col=2,
     ylim=c(min,max),xlab="",ylab="")
points(1:m,interval.known[,2],pch=19,col=2)
points(1:m,interval.unknown[,1])
points(1:m,interval.unknown[,2])
segments(1:m,interval.known[,1],1:m,interval.known[,2])
segments(1:m,interval.unknown[,1],1:m,interval.unknown[,2])
abline(h=2,col=3)
```



```
mean(interval.known<interval.unknown)
```

```
[1] 0.5216667
```