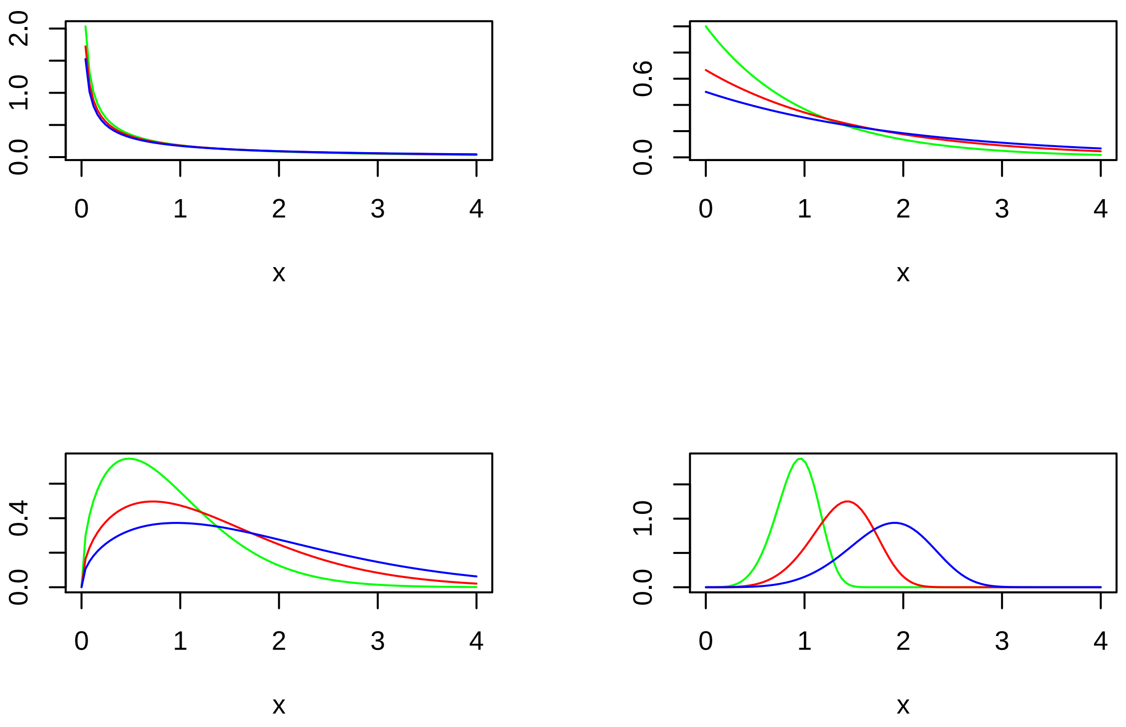
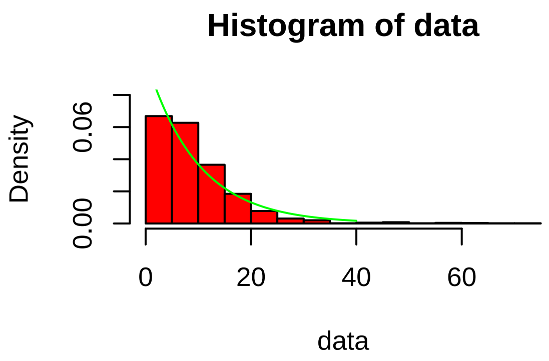
alley

Andreas Dahlberg

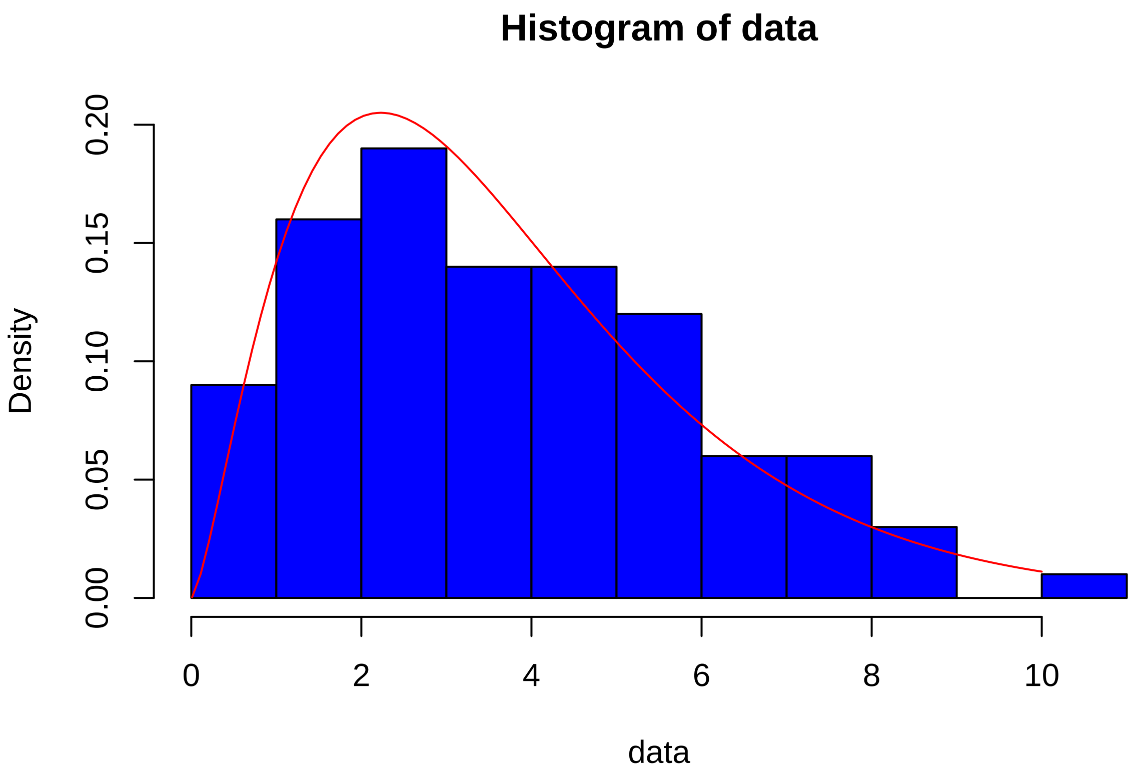
```r
x = seq(0,4,length = 100)
s = c(1,1.5,2)
k=c(0.5,1.0,1.5,5.0)
par(mfrow = c(2,2))
cols = c("GREEN","RED","BLUE")
for(i in 1:4){
  for(j in 1:3){
    if (j == 1)
      plot(x,dweibull(x,k[i],s[j]),col=cols[j],type="l",ylab="",xlab="x")
    else
      lines(x,dweibull(x,k[i],s[j]),col=cols[j],type="l",ylab="",xlab="x")
  }
}
```



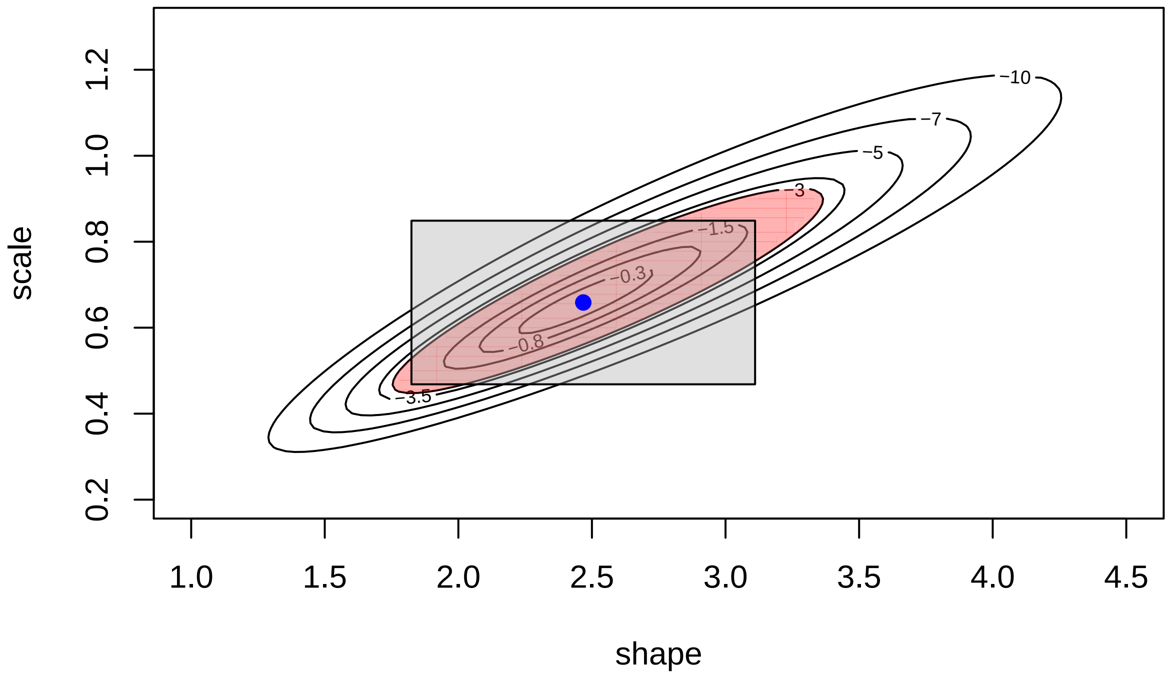
```r
data = read.csv("hospital.csv")$stay
hist(data,prob="TRUE",col="red",ylim=c(0,0.08))

log_likeli <- function(l){
  sum(dexp(data,l,log=TRUE))
}
opt = optim(1,function(l)-log_likeli(l),method="Brent",lower=0,upper=50)$par

x <- seq(0,40,length=100)
lines(x,dexp(x,opt),col="green",type="l")
```



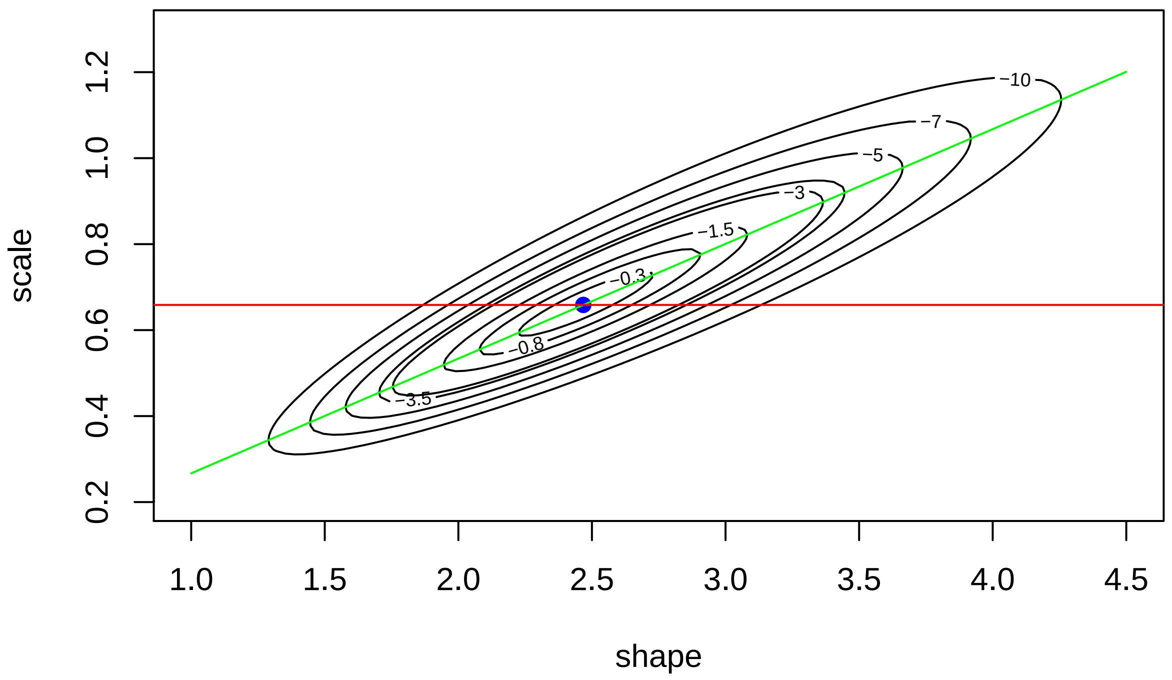
```
## CTRL+SHIFT+c f??r kommentera flera lines
data = get(load("exercise2.RData"))
hist(data,prob="TRUE",col="BLUE",breaks=10,ylim=c(0,0.20))

log_likeli <- function(x){
  sum(dgamma(data,x[1],x[2],log=TRUE))
}
opt = optim(c(2.5,0.65),function(x)-log_likeli(x),hessian=TRUE)
ML = opt$par
max=log_likeli(ML)
obs_fisher = opt$hessian
se = sqrt(diag(solve(obs_fisher)))

wald_shape = ML[1]+c(-1,1)*1.96*se[1]
wald_scale = ML[2]+c(-1,1)*1.96*se[2]

x = seq(0.01,10,length=100)
lines(x,dgamma(x,ML[1],ML[2]),col="red")
```



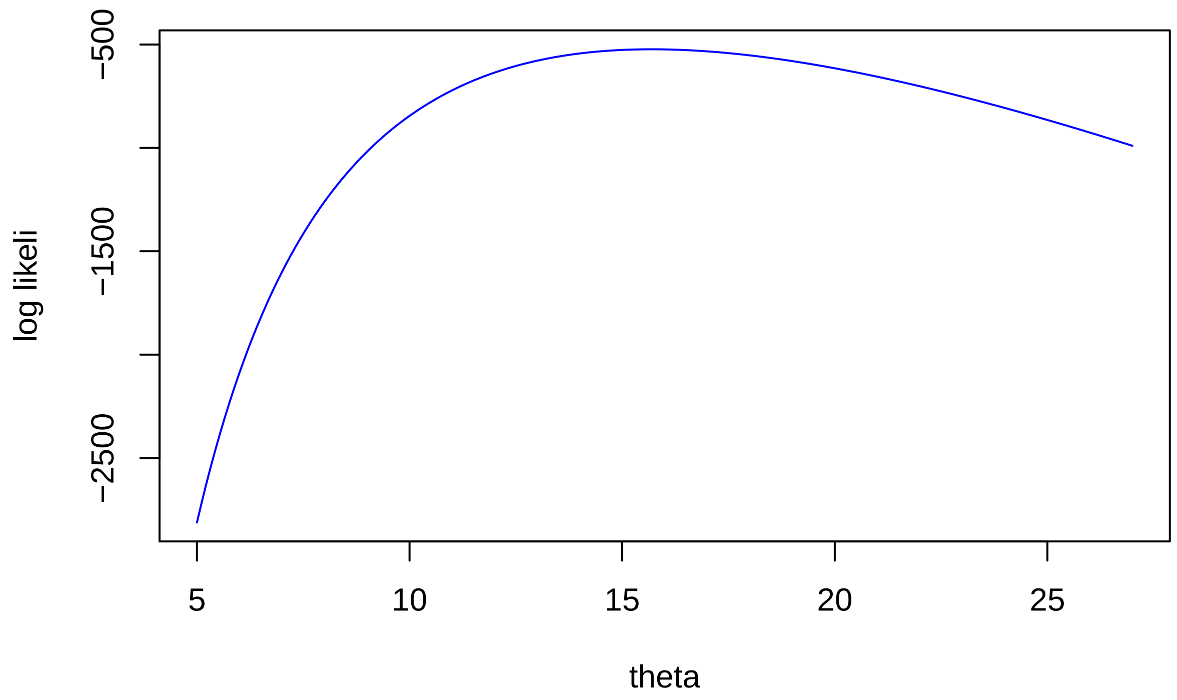
```
rel_log_likeli <- function(x){
  log_likeli(x)-max
}

shape = seq(1,4.5,length=100)
scale = seq(0.2,1.3,length=100)
grid = expand.grid(shape,scale)
z = apply(grid,1,rel_log_likeli) ## kan ocksa anvanda outer(shape,scale,rel_log_likeli) men maste da va
z = array(z,c(100,100))
contour(shape,scale,z,levels=c(0,-0.3,-0.8,-1.5,-3.0,-3.5,-5,-7,-10),xlab="shape",ylab="scale")
.filled.contour(shape,scale,z,levels=c(-qchisq(0.95,2)/2,0),col=rgb(1, 0, 0, 0.3)) #likelihood ratio CI
rect(xleft=wald_shape[1],xright=wald_shape[2],ybottom=wald_scale[1],ytop=wald_scale[2],col = rgb(0.7, 0
points(ML[1],ML[2],pch=19,col="BLUE")
```



```
contour(shape,scale,z,levels=c(0,-0.3,-0.8,-1.5,-3.0,-3.5,-5,-7,-10),xlab="shape",ylab="scale")
points(ML[1],ML[2],pch=19,col="BLUE")

prof_log_likeli_shape <- function(shape){
  opt = optim(0.6,function(scale)-log_likeli(c(shape,scale)),method="Brent",lower=0,upper=50) #om ger N
  opt$par
}
lines(shape,sapply(shape,prof_log_likeli_shape),col="GREEN",type="l")
abline(h=ML[2],col="red")
```

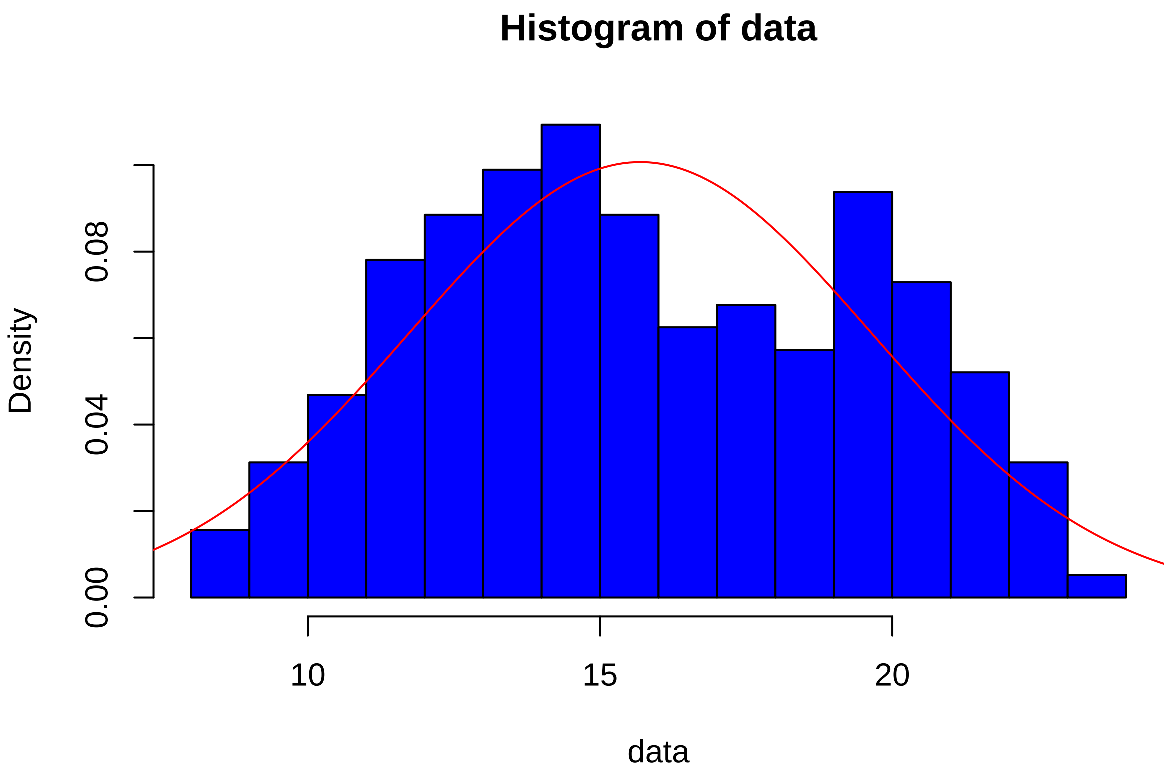


```r
data = read.csv("zurichtemp.csv")$Temperature
log_likeli <- function(theta){
  sum(dnorm(data,theta,sqrt(theta),log=TRUE))
}
theta = seq(5,27,length=200)
plot(theta,sapply(theta,log_likeli),col="blue",type="l",ylab="log likeli")
```

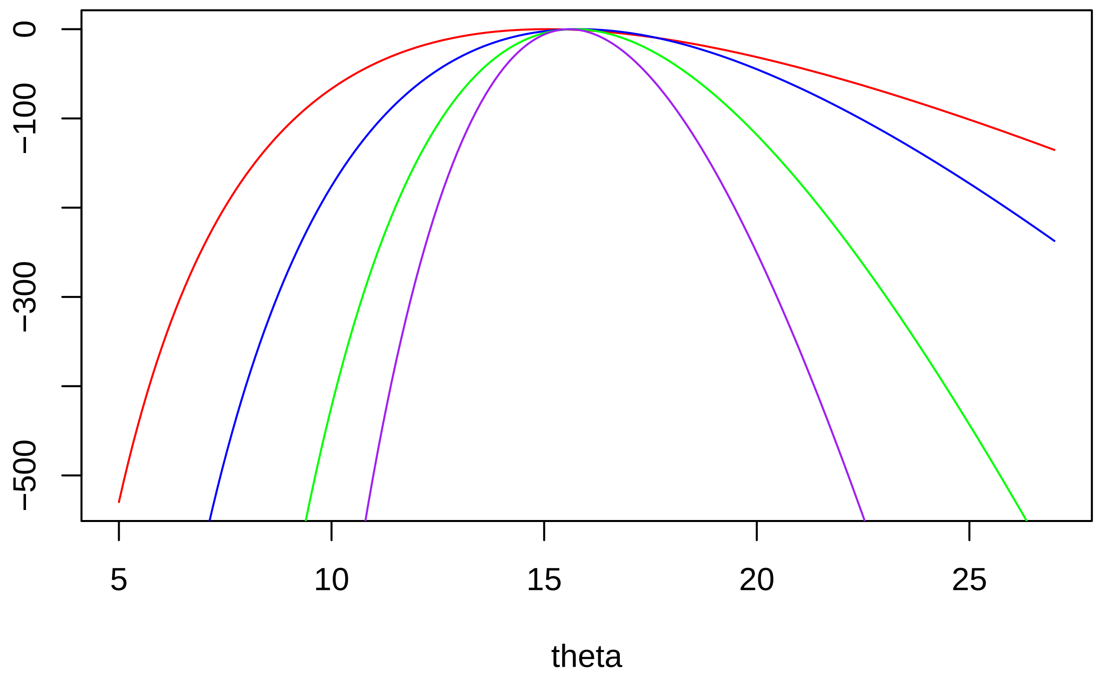


```r
opt = optim(18,function(theta)-log_likeli(theta),method="Brent",lower=0,upper=50)
ML = opt$par

hist(data,prob="TRUE",col="blue",breaks=15)
```

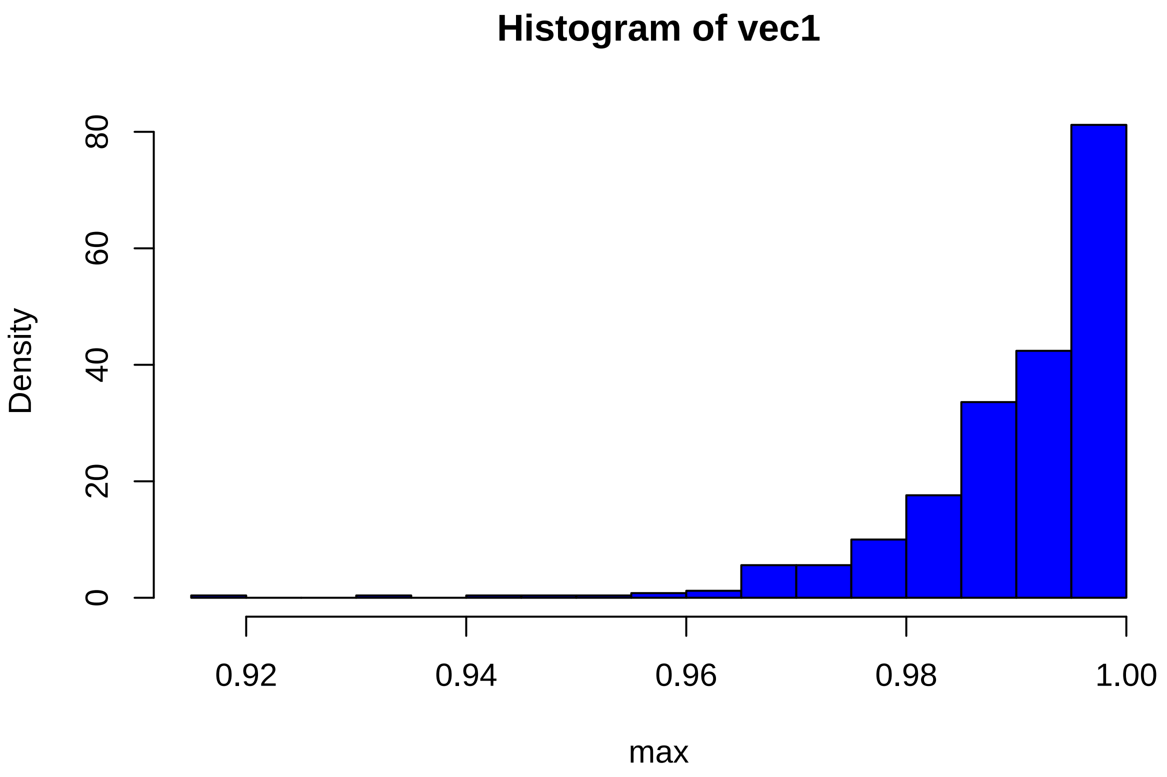
```r
lines(theta,dnorm(theta,ML,sqrt(ML)),col="red",type="l")
```



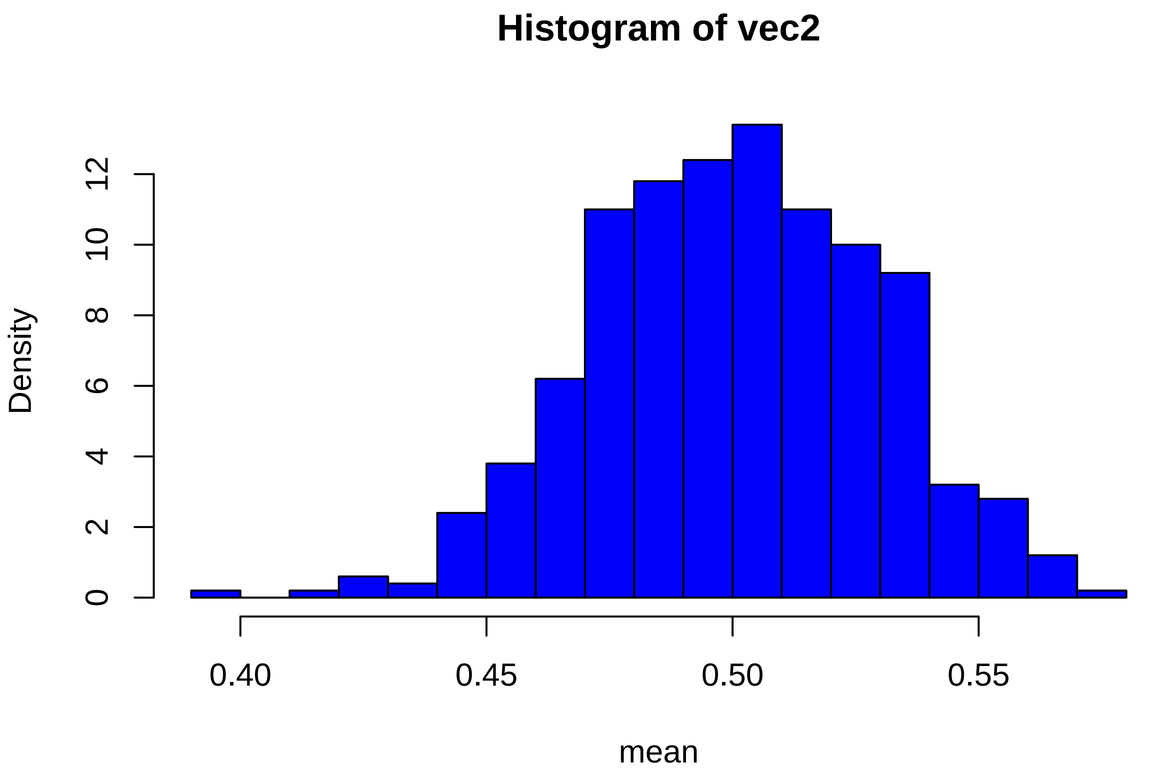
```r
sizes = c(50,100,250,500)
cols=c("RED","BLUE","GREEN","PURPLE")
for (i in 1:4){
  sample = rnorm(sizes[i],ML,sqrt(ML))
  log_likeli2 <- function(theta){
    sum(dnorm(sample,theta,sqrt(theta),log=TRUE))
  }
  ML2 = optim(18,function(theta)-log_likeli2(theta),method="Brent",lower=0,upper=50)$par
  max = log_likeli2(ML2)
  if (i==1)
    plot(theta,sapply(theta,function(theta) log_likeli2(theta)-max),col=cols[i],type="l",ylab="")
  else
    lines(theta,sapply(theta,function(theta) log_likeli2(theta)-max),col=cols[i],type="l")

}
```



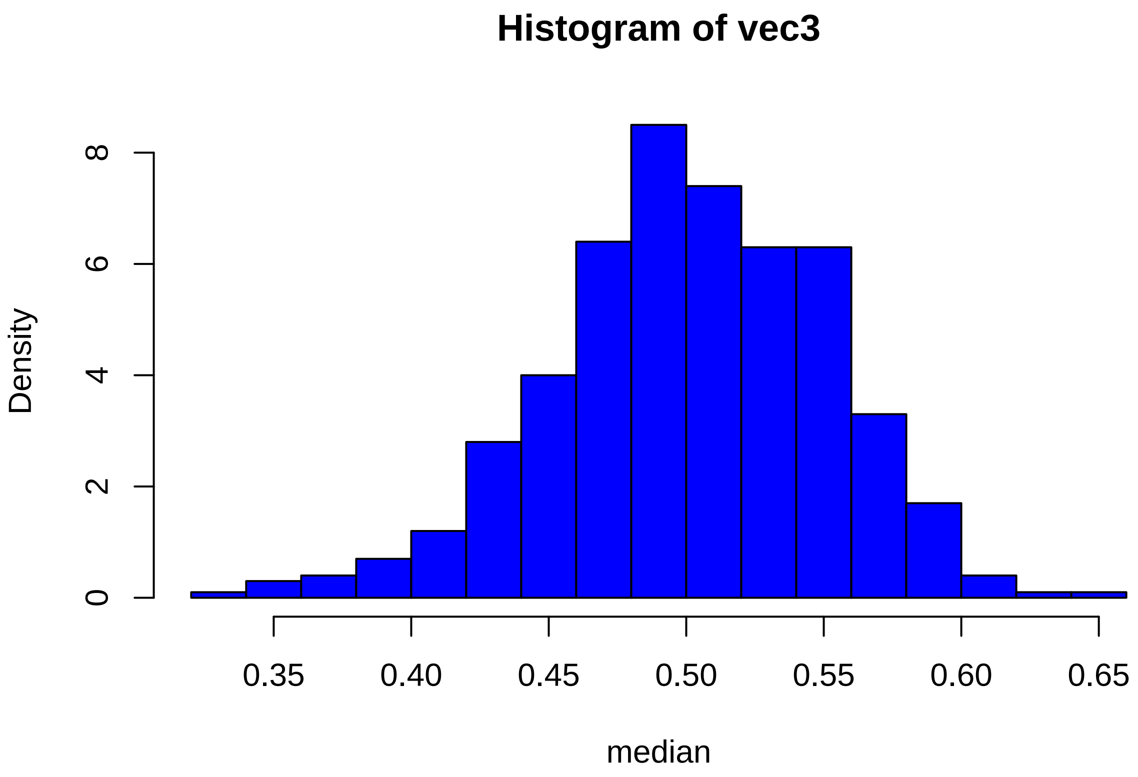
```r
vec1 = c()
vec2 = c()
vec3 = c()
vec4 = c()
for (j in 1:500){
  sample = runif(100,0,1)
  vec1 = append(vec1,max(sample))
  vec2 = append(vec2,mean(sample))
  vec3 = append(vec3,median(sample))
  vec4 = append(vec4,min(sample))
}
hist(vec1,col="blue",prob="TRUE",xlab="max",breaks=20)
```



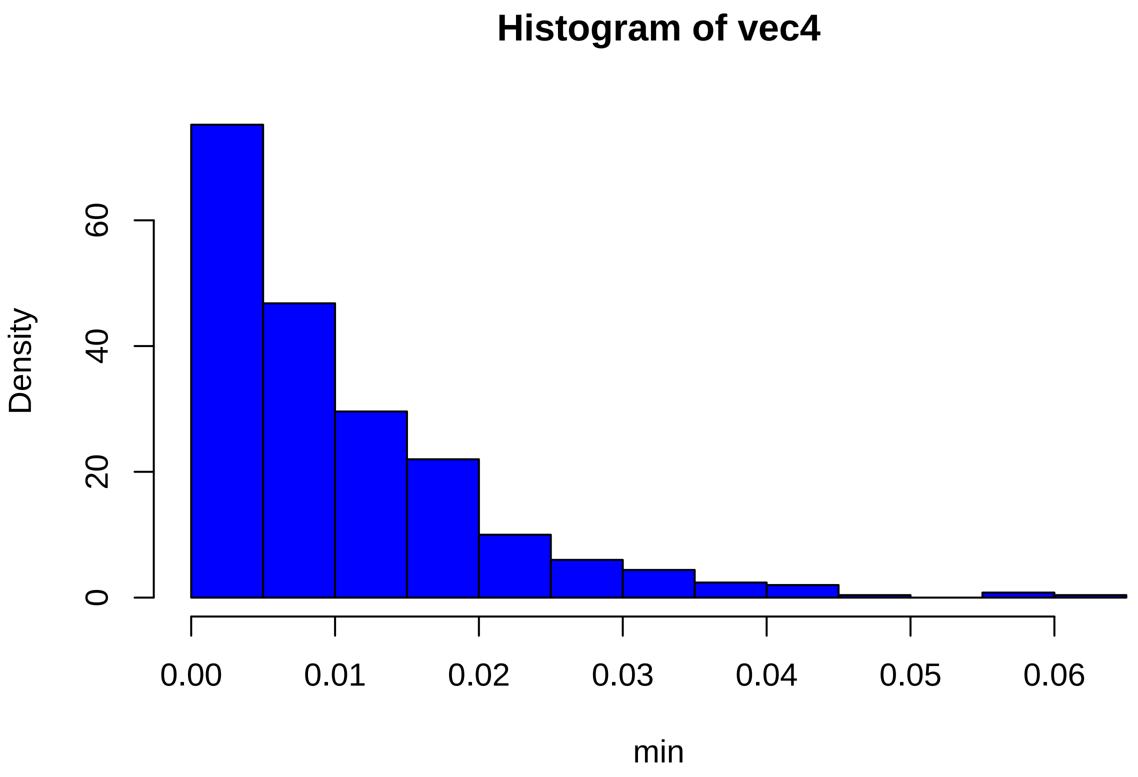
```r
hist(vec2,col="blue",prob="TRUE",xlab="mean",breaks=20)
```



```r
hist(vec3,col="blue",prob="TRUE",xlab="median",breaks=20)
```

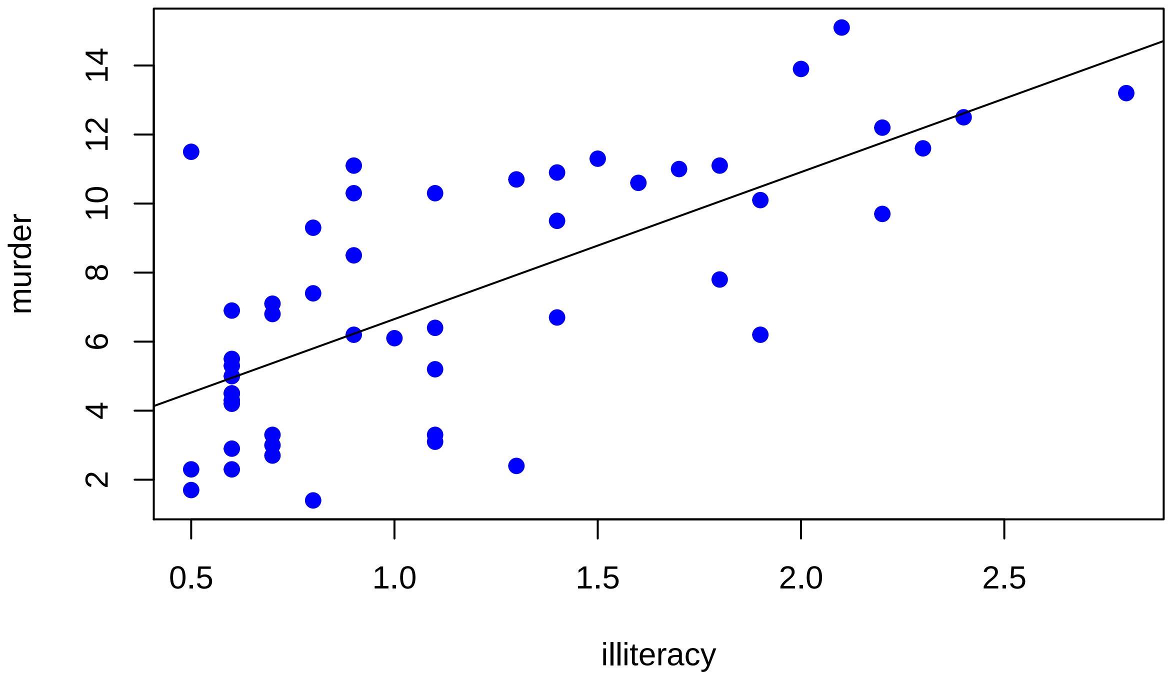


```r
hist(vec4,col="blue",prob="TRUE",xlab="min",breaks=20)
```

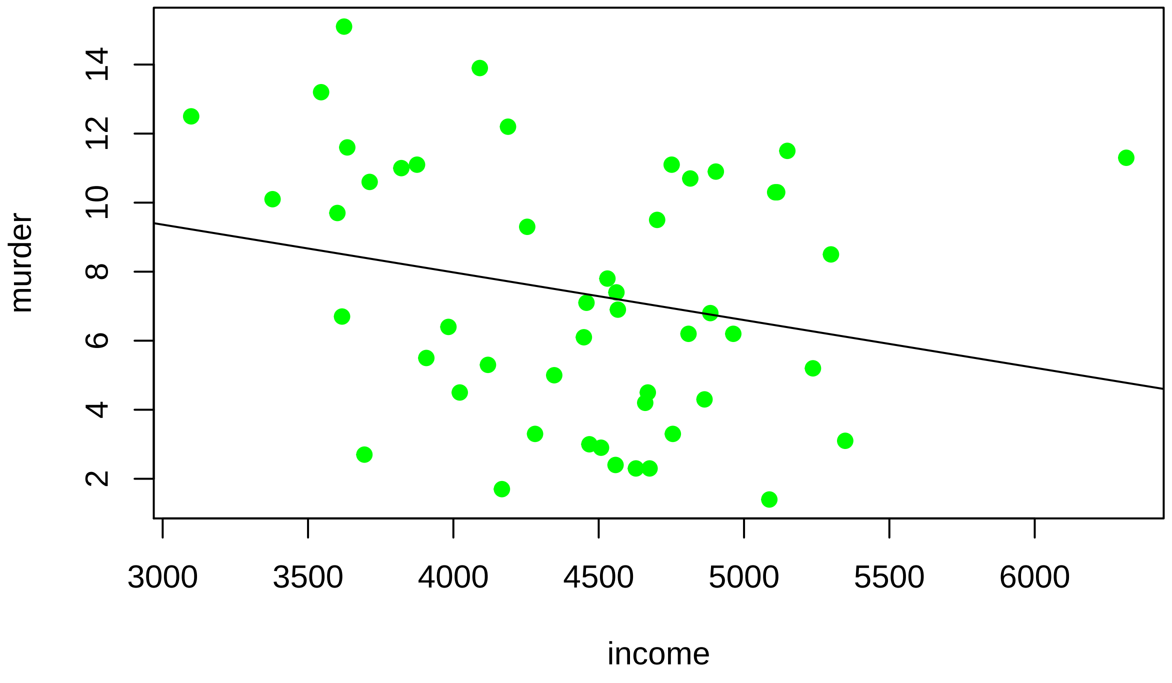


```r
library(MASS)
states <- as.data.frame(state.x77[, c("Murder", "Population", "Illiteracy", "Income", "Frost")])
murder = states$Murder
population = states$Population
```

```r
illiteracy = states$Illiteracy
income = states$Income
frost = states$Frost

plot(illiteracy,murder,pch=19,col="blue",type="p") ##kan v??lja linje eller pkt h??r
lm1 = lm(murder~illiteracy)
abline(lm1)
```



```r
plot(income,murder,pch=19,col="green",type="p")
lm2 = lm(murder~income)
abline(lm2)
```



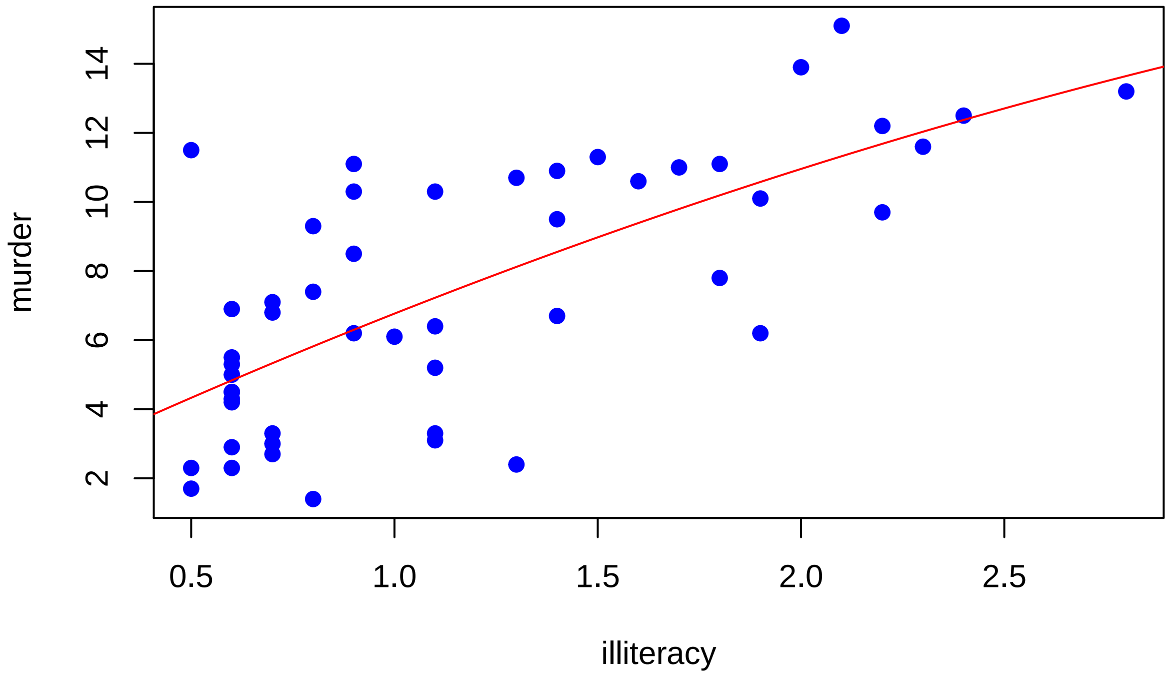
```
confint(lm1,level=0.95)
```

```
##                  2.5 %   97.5 %
## (Intercept) 0.7511819 4.042369
## illiteracy  3.0074184 5.507495
```

```
confint(lm2,level=0.95)
```

```
##                     2.5 %       97.5 %
## (Intercept)  5.912577550 2.110604e+01
## income      -0.003078949 3.144832e-04
```

```
illiteracy2 = illiteracy^2
lm3 = lm(murder ~ illiteracy+illiteracy2)
coeffs = (summary(lm3)$coefficients)[,1]
fun <- function(x){
  coeffs[1]+coeffs[2]*x+coeffs[3]*x^2
}
plot(illiteracy,murder,pch=19,col="blue",type="p")
x = seq(0,3,length=100)
lines(x,fun(x),col="red",type="l")
```



```
confint(lm3,level=0.95)
```

```
##                  2.5 %    97.5 %
## (Intercept) -2.1995046  5.524955
## illiteracy  -0.7700334 11.898412
## illiteracy2 -2.6373110  1.720027
```

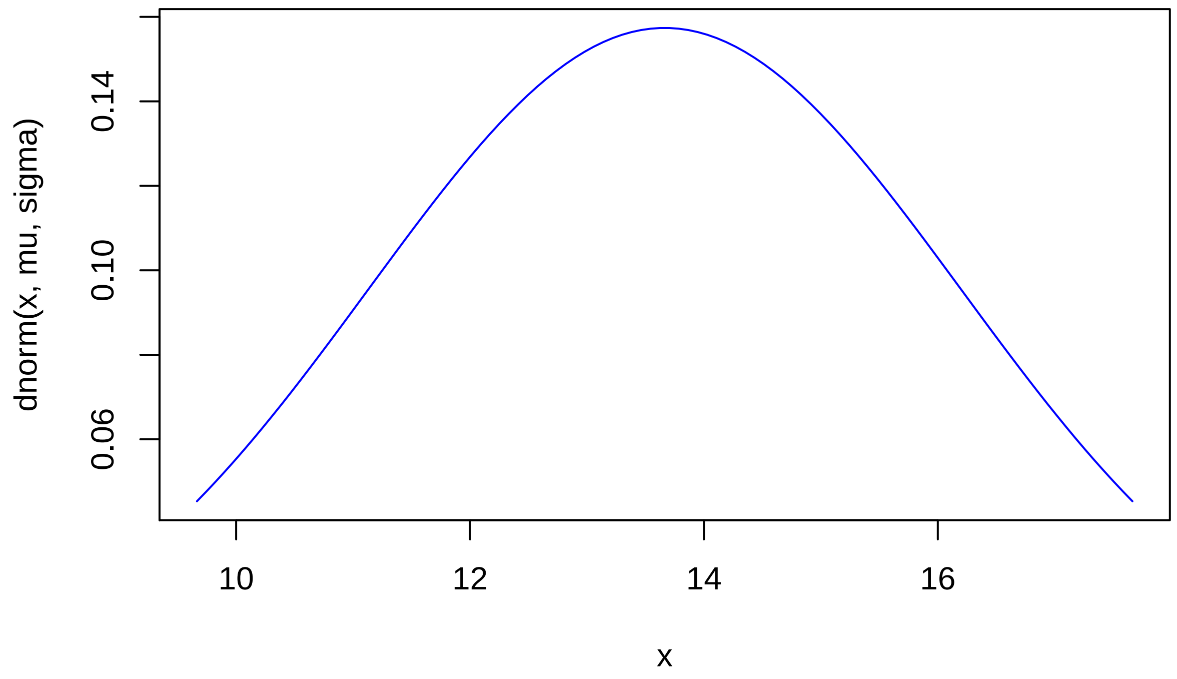
```
lm = lm(murder ~ population+illiteracy+income+frost)
coeffs = (summary(lm)$coefficients)[,1]
confint(lm,level=0.95)
```

```
##                     2.5 %       97.5 %
## (Intercept) -6.552191e+00 9.0213182149
## population   4.136397e-05 0.0004059867
## illiteracy   2.381799e+00 5.9038743192
## income      -1.312611e-03 0.0014414600
## frost       -1.966781e-02 0.0208304170
```

```
sigma = (summary(lm)$sigma)
new_point = c(1,2,3,1,1.7)
mu = new_point%*%coeffs
x = seq(mu-4,mu+4,length=100)
```

```
## Warning in seq_len(length.out - 2L) * by: Recycling array of length 1 in vector-array arithmetic is 
##   Use c() or as.vector() instead.

## Warning in from + seq_len(length.out - 2L) * by: Recycling array of length 1 in array-vector arithme
##   Use c() or as.vector() instead.
```

```
plot(x,dnorm(x,mu,sigma),col="BLUE",type="l") ##plug in
```



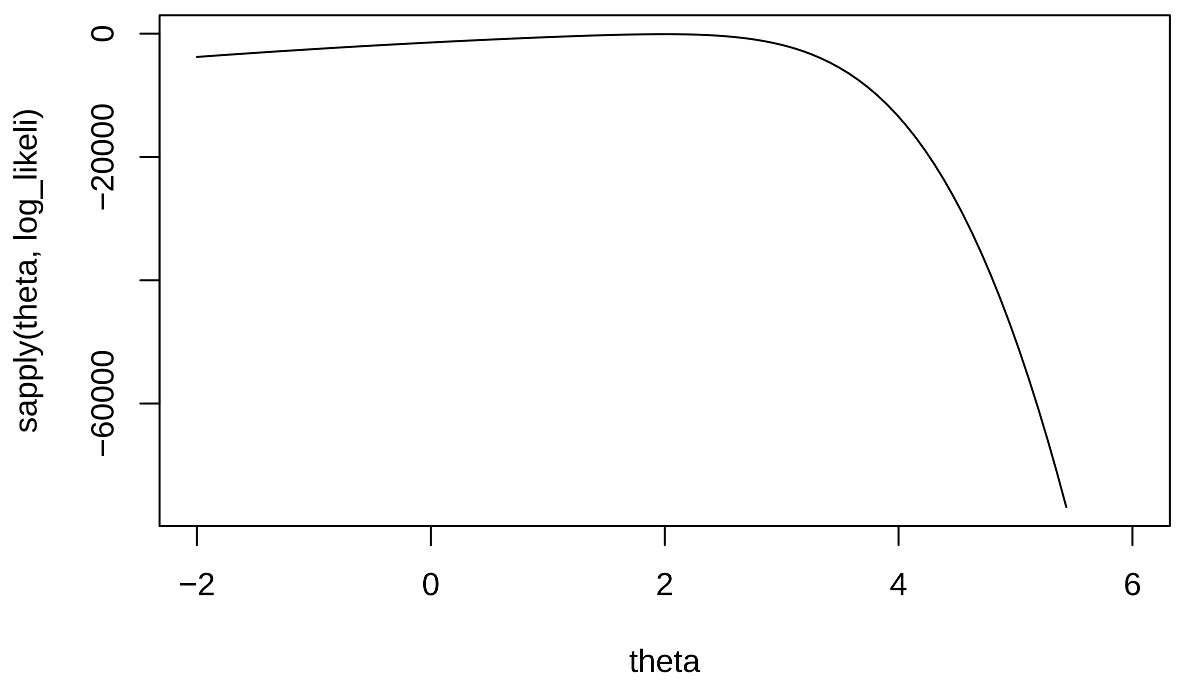
```
k = stepAIC(lm,direction="backward")
```

```
## Start:  AIC=97.75
## murder ~ population + illiteracy + income + frost
##
##              Df Sum of Sq    RSS     AIC
## - frost       1     0.021 289.19  95.753
## - income      1     0.057 289.22  95.759
## <none>                    289.17  97.749
## - population  1    39.238 328.41 102.111
## - illiteracy  1   144.264 433.43 115.986
##
## Step:  AIC=95.75
## murder ~ population + illiteracy + income
##
##              Df Sum of Sq    RSS     AIC
## - income      1     0.057 289.25  93.763
## <none>                      289.19  95.753
## - population  1    43.658 332.85 100.783
## - illiteracy  1   236.196 525.38 123.605
##
## Step:  AIC=93.76
## murder ~ population + illiteracy
##
##              Df Sum of Sq    RSS     AIC
## <none>                      289.25  93.763
## - population  1    48.517 337.76  99.516
## - illiteracy  1   299.646 588.89 127.311
```

```r
k
```

```
##
## Call:
## lm(formula = murder ~ population + illiteracy)
##
## Coefficients:
## (Intercept)   population   illiteracy
##   1.6515497    0.0002242    4.0807366
```

```r
vec = c()
for (i in 1:200){
  sample = rnorm(300,2,1)
  vec = append(vec,max(sample))
}

pdf <- function(y,theta){
  300*pnorm(y-theta)^299*dnorm(y-theta)
}

log_likeli <- function(theta){
  sum(sapply(vec,function(x) log(pdf(x,theta))))
}

theta = seq(-2,6,length=100)
plot(theta,sapply(theta,log_likeli),type="l")
```



```
ML = optim(4,function(theta)-log_likeli(theta),method="BFGS")$par
ML
```

```
## [1] 1.997302
```

```
c = 3.04
favorable = 0
for (i in 1:10000){
  x = rpois(1,2*c)
  CI = 1/c*(x+c(-1,1)*1.96*sqrt(x))
  if (2 >= CI[1] && 2 <= CI[2])
    favorable = favorable+1
}
favorable/10000
```

```
## [1] 0.9345
```

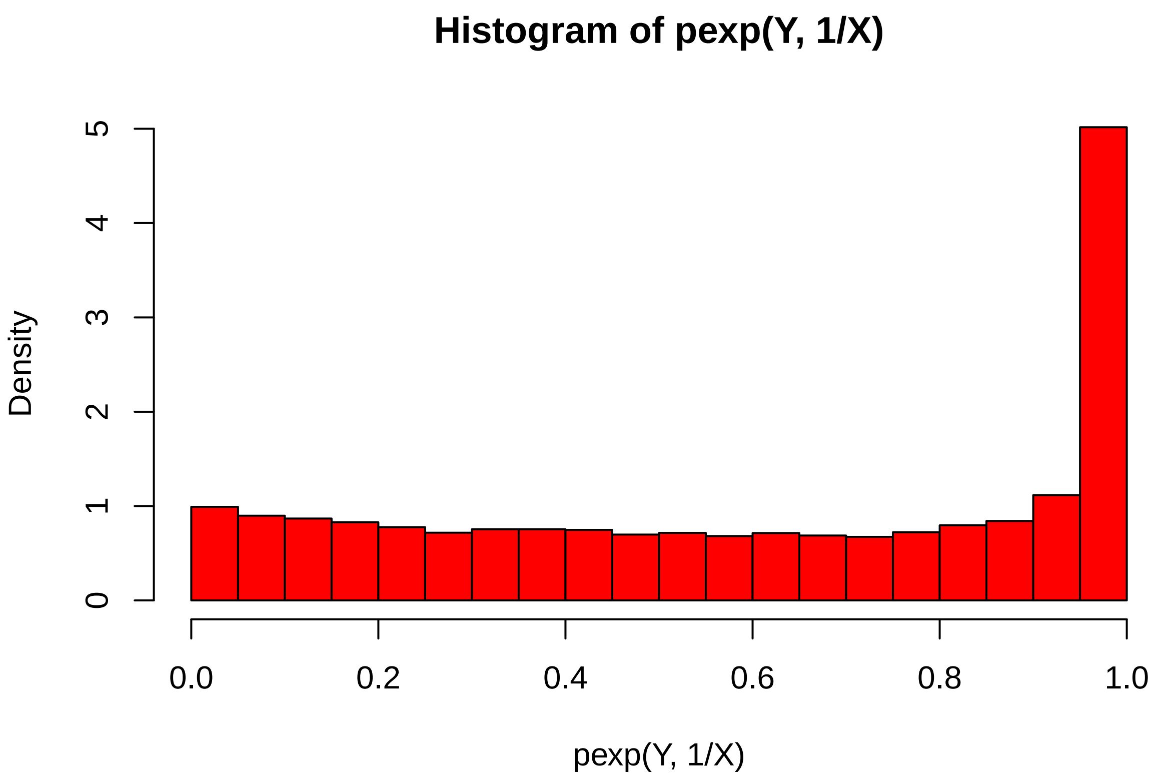
```
n=100
x=2
fun <- function(y){
  exp(y)/(1+exp(y))
}
fun(log(x/(n-x))+c(-1,1)*1.96*sqrt(n/(x*(n-x))))
```

```
## [1] 0.005007391 0.076433601
```

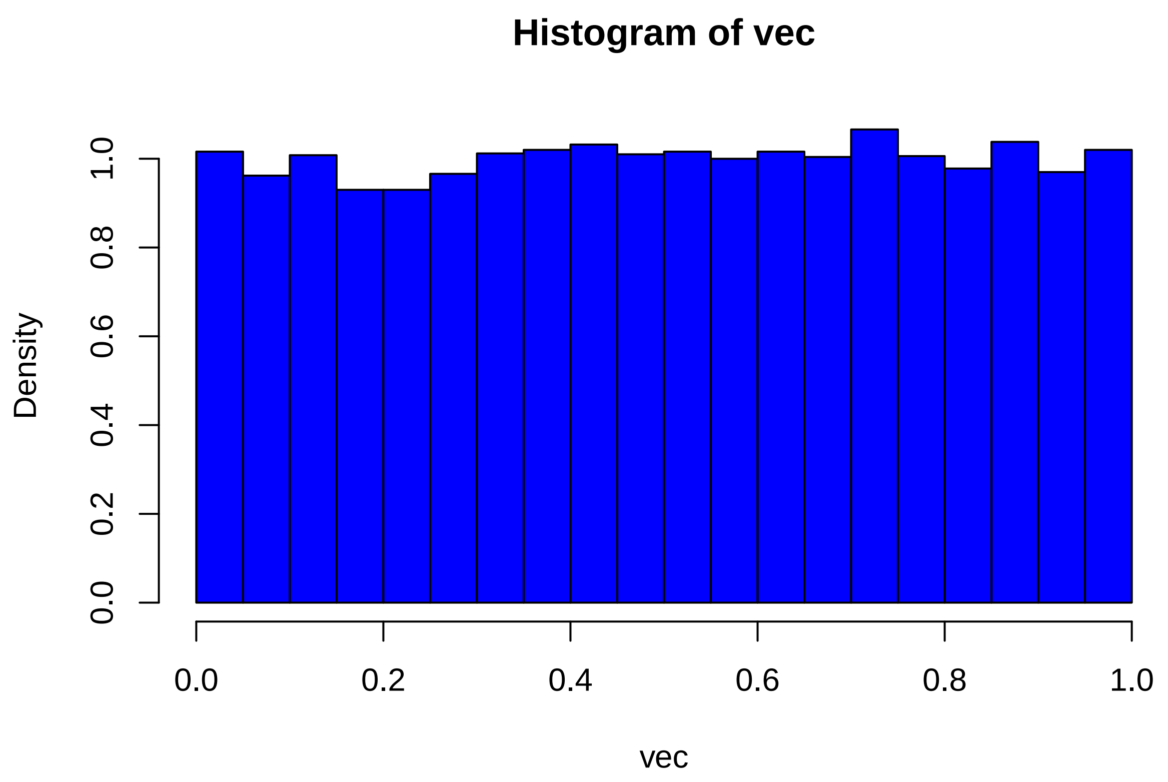
```
x1=6
n1=108
x2=3
n2=103

2*log( ( (x1/n1)^x1*(1-x1/n1)^(n1-x1)*(x2/n2)^x2*(1-x2/n2)^(n2-x2))/(((x1+x2)/(n1+n2))^(x1+x2)*(1-(x1+
```

```
## [1] 0.8351912
```

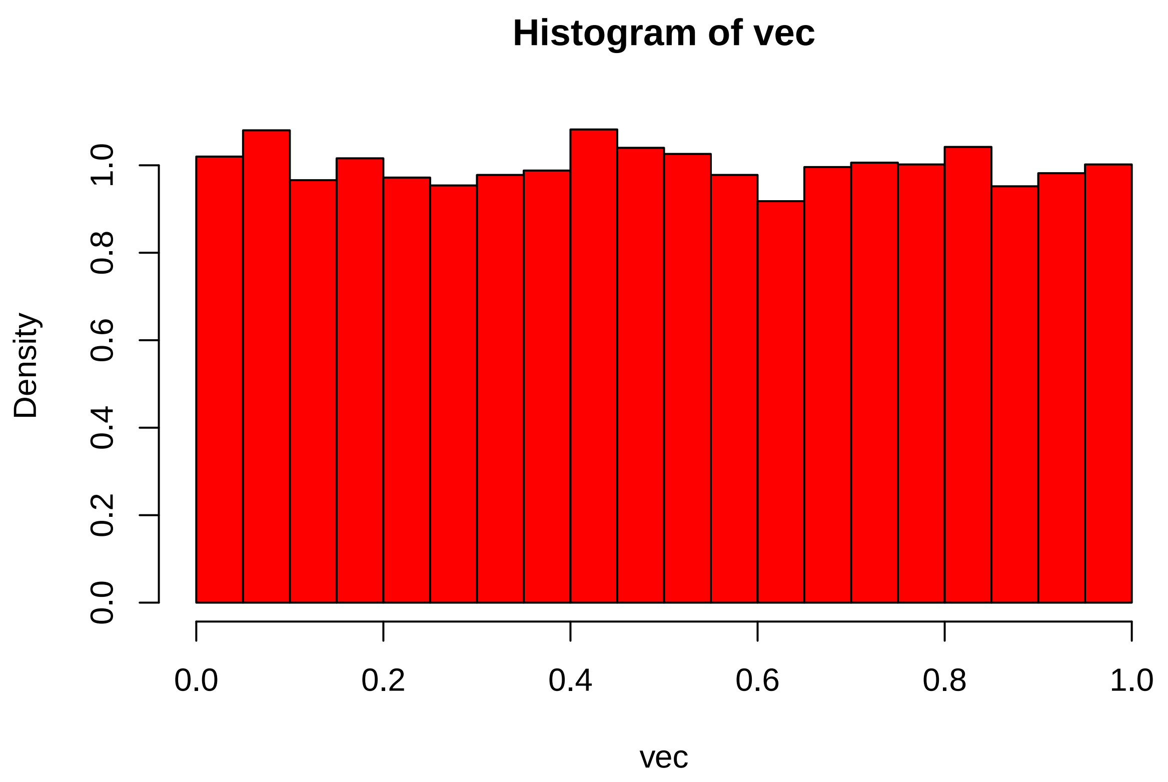
```r
X = rexp(10000,2)
Y = rexp(10000,2)
hist(pexp(Y,1/X),prob="TRUE",col="red")
```



```r
pdf = function(x,y){
  x/(x+y)^2
}
cdf = function(x,y){
  integrate(function(y)pdf(x,y),0,y)$value
}

vec = c()
for(i in 1:length(X)){
  vec = append(vec,cdf(X[i],Y[i]))
}
hist(vec,prob="true",col="blue")
```



```
lambda = rgamma(10000,2,2) ##slumpa fram riktiga grejjer, och ta F med din predictive distr!
X = rexp(10000,lambda)
Y = rexp(10000,lambda)
pdf = function(x,y){ ##alpha=2,beta=2
  3*(x+2)^3/(x+y+2)^4
}
cdf = function(x,y){
  integrate(function(y)pdf(x,y),0,y)$value
}
vec = c()
for(i in 1:length(X)){
  vec = append(vec,cdf(X[i],Y[i]))
}
hist(vec,prob="TRUE",col="red")
```

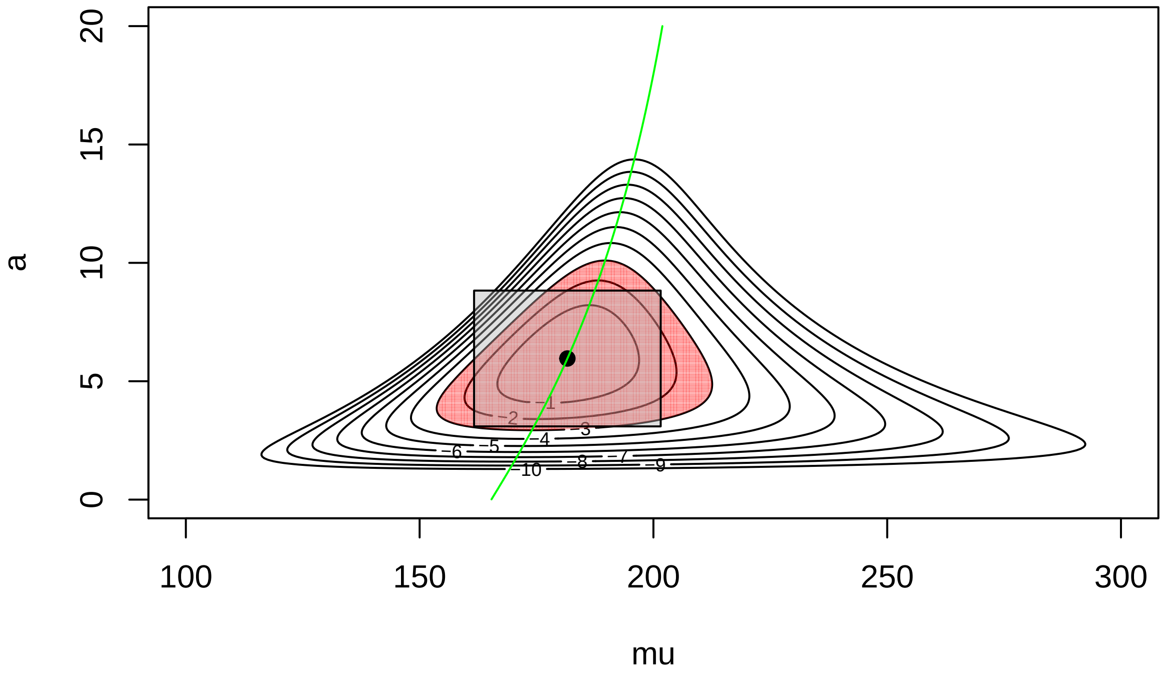


```
data = c(225, 175, 198, 189, 189, 130, 162, 135, 119, 162)
n = length(data)

log_likeli <- function(x){
  mu = x[1]
  a = x[2]
  n*log(a)-a*n*log(mu)+(a-1)*sum(log(data))-1/(mu^a)*sum(data^a)
}
opt = optim(c(180,2),function(x)-log_likeli(x),hessian=TRUE)
```

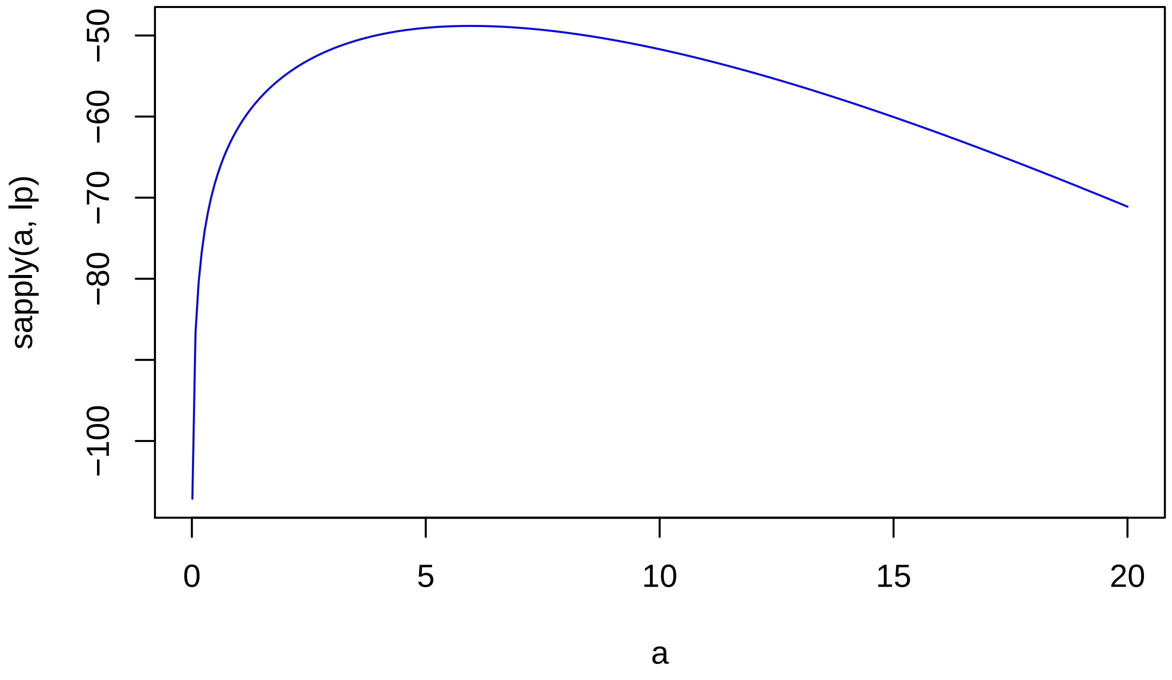
```
## Warning in log(a): NaNs produced

## Warning in log(a): NaNs produced
```

```
observed_fisher=opt$hessian
ML = opt$par
se = sqrt(diag(solve(observed_fisher)))
max = -opt$val
wald_mu = ML[1]+c(-1,1)*1.96*se[1]
wald_a = ML[2]+c(-1,1)*1.96*se[2]


mu = seq(100,300,length=300)
a = seq(0.01,20,length=300)
grid = expand.grid(mu,a)
z = apply(grid,1,function(x)log_likeli(x)-max)
z = array(z,c(300,300))
contour(mu,a,z,levels=0:-10,xlab="mu",ylab="a")
.filled.contour(mu,a,z,levels=c(-qchisq(0.95,2)/2,0),col=rgb(1, 0, 0, 0.3))
rect(xleft=wald_mu[1],xright=wald_mu[2],ybottom=wald_a[1],ytop=wald_a[2],col = rgb(0.7, 0.7, 0.7, 0.4))
```

```r
points(ML[1],ML[2],pch=19,col="black")

profile_log_likeli <- function(a){
  opt = optim(150,function(mu)-log_likeli(c(mu,a)),method="Brent",lower=0,upper=1000)$par
}
lines(sapply(a,profile_log_likeli),a,col="green",type="l")
```



```r
lp <- function(a){
  n*log(a)-n*log(sum(data^a)/n)+(a-1)*sum(log(data))-n
}
plot(a,sapply(a,lp),type="l",col="blue")
```



```r
a = 8.78
b = 8.78
x = 41
c = 23.3

lambda = seq(0.1,10,length=200)
plot(lambda,dgamma(lambda,a,b),col="blue",type="l",ylim=c(0,2))
lines(lambda,dgamma(lambda,a+x,b+c),col="green",type="l")

lambda = 0:10
lines(lambda,dpois(x,c*lambda)*40,type="l",col="red")
legend("topright",legend=c("prior","posterior","likelihood"),col=c("blue","green","red"),lty = rep(1,3)
## ser att det ??r en kombination av prior och likelihood som ger posterior

posterior <- function(l){
  dgamma(l,a+x,b+c)
}

mode = optim(1,function(l)-posterior(l),method="BFGS")$par
mode
```

```
## [1] 1.52057
```

```r
mean = integrate(function(l) l*posterior(l),0,Inf)$val
mean
```

```
## [1] 1.551746
```

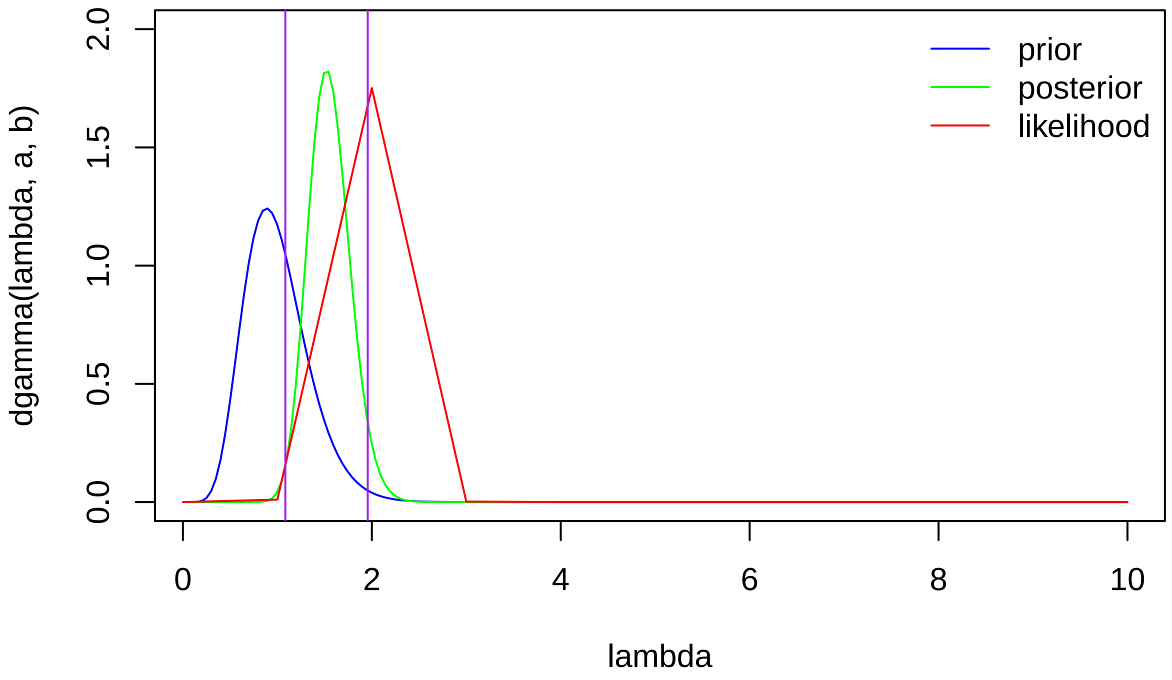
```r
to_min <- function(m){
  abs(integrate(function(l) posterior(l),0,m)$val - 0.5)
}
median = optim(1,to_min,method="BFGS")$par
median
```

```
## [1] 1.541368
```

```r
to_min <- function(eps){

  abs(integrate(function(l) posterior(l),mode-eps,mode+eps)$val - 0.95)
}
eps = optim(0.1,to_min,method="Brent",lower=0,upper=1)$par
ci = c(mode-eps,mode+eps)
ci
```

```
## [1] 1.084562 1.956578
```

```r
abline(v=ci[1],col="purple")
abline(v=ci[2],col="purple")
```


prior
posterior
likelihood
dgamma(lambda, a, b)
lambda
0.0
0.5
1.0
1.5
2.0
0
2
4
6
8
10