Lab 6: Likelihood

36-350, Statistical Computing

SOLUTIONS

Load the data and the method-of-moments estimator:

```
> library(MASS)
> data(cats)
> gamma.est <- function(data) {
+  m <- mean(data)
+  v <- var(data)
+  s <- v/m
+  a <- m/s
+  fit <- c(a,s)
+  names(fit) <- c("shape","scale")
+  return(fit)
+ }
```

1. Solution:

   ```
   > (fit.mm <- gamma.est(cats$Hwt))
        shape      scale
   19.0653121  0.5575862
   ```

   The assignment operator `<-` invisibly returns the value it assigns. By putting the whole assignment in parentheses, we get it echoed to the screen.

2. Solution:

   ```
   > sum(dgamma(x=cats$Hwt,shape=fit.mm["shape"],scale=fit.mm["scale"],log=TRUE))
   [1] -325.6886
   ```

   Because of the importance of the log-likelihood, most density functions include an option for directly returning the logs of probability densities. In fact, for many distributions, R internally calculates the log density first (because it's easier), and then exponentiates, so using the log option is actually slightly more accurate numerically.

3. Solution: This resembles things like `make.linear.predictor` in Lecture 11.

```
make.gamma.loglike <- function(x) {
  loglike <- function(theta) {
    return(sum(dgamma(x,shape=theta[1],scale=theta[2],log=TRUE)))
  }
  return(loglike)
}
```

The check:

```
> cats.gamma.loglike <- make.gamma.loglike(cats$Hwt)
> cats.gamma.loglike(fit.mm)
[1] -325.6886
```

4. SOLUTION: Using the hint,

```
surface.0(cats.gamma.loglike,from.x=1,to.x=40,from.y=0.01,to.y=1,nlevels=100,
  xlab="shape",ylab="scale")
points(x=fit.mm["shape"],y=fit.mm["scale"],col="red")
```

The result is Figure 1.

5. SOLUTION: Notice that the contour lines of log-likelihood all have negative values. The two highest contour values, both -2000, are on the two visually-uppermost curves, which bracket the method-of-moments estimate, which we know is at ≈ -326. We therefore zoom in on this vicinity:

```
surface.0(cats.gamma.loglike,from.x=15,to.x=25,from.y=0.5,to.y=.7,nlevels=100,
  xlab="shape",ylab="scale")
points(x=fit.mm["shape"],y=fit.mm["scale"],col="red")
```

The result, in Figure 2, still has the method-of-moments estimate bracketed between the two highest contour lines, though now their log-likelihood is -330.

6. SOLUTION: Gradient descent *minimizes* a function, but we want to *maximize* the log-likelihood. We have two options. One is to define a new function, which is just the negative log-likelihood. The other is to notice that if we give `gradient.descent` negative numbers in the `step.scale` argument, it will in fact adjust the parameter estimate in the direction of the gradient instead of against it, and so move towards a local maximum.

After some playing with the control settings, and (on my computer) seven seconds of run time,

```
> (fit.ml <- gradient.descent(f=cats.gamma.loglike,initial.x=fit.mm,
+  max.iterations=1e4,step.scale=-c(1e-2,1e-4),stopping.deriv=1e-4))
```

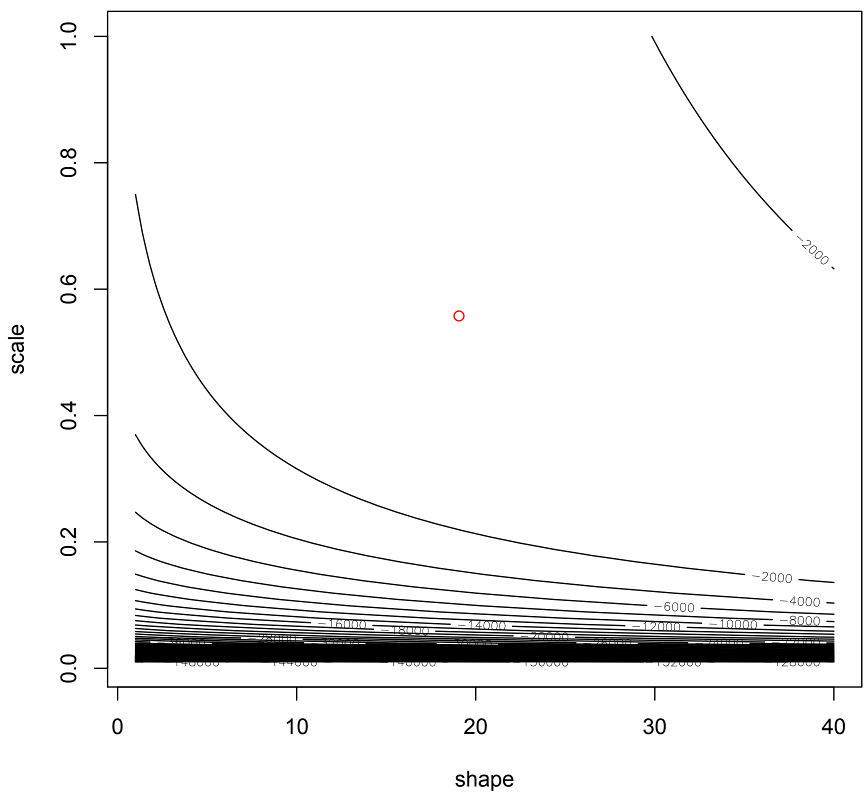


Figure 1: Contour lines of log likelihood for the cats' hearts' weight data and the gamma distribution model. The red dot makes the method-of-moments estimate from Lab 3.

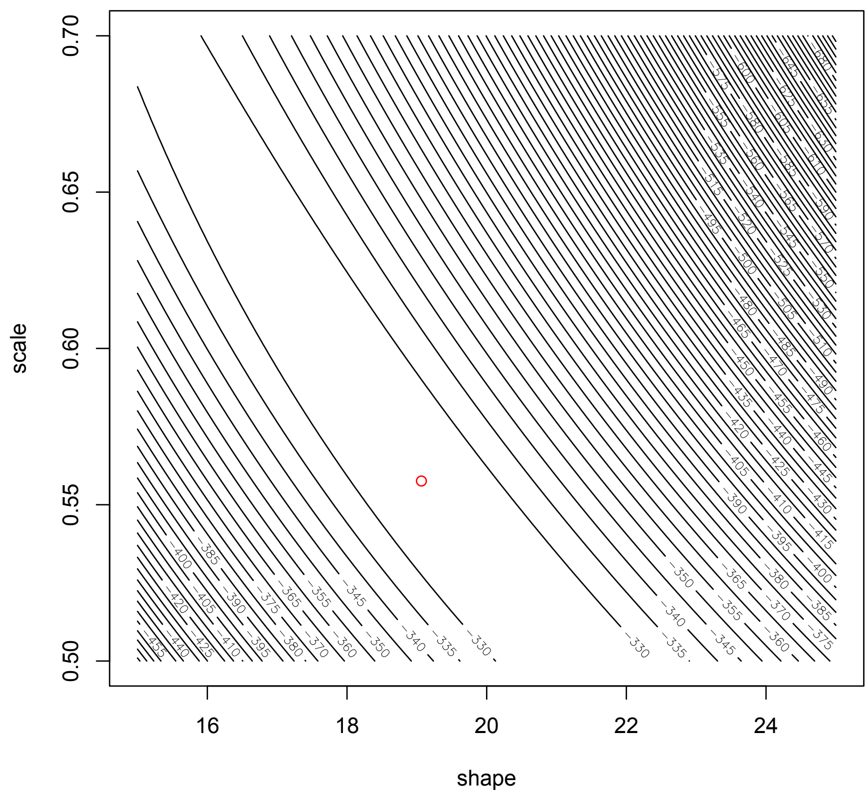


Figure 2: Zoom-in on the region of parameters which seem to have the highest likelihood.

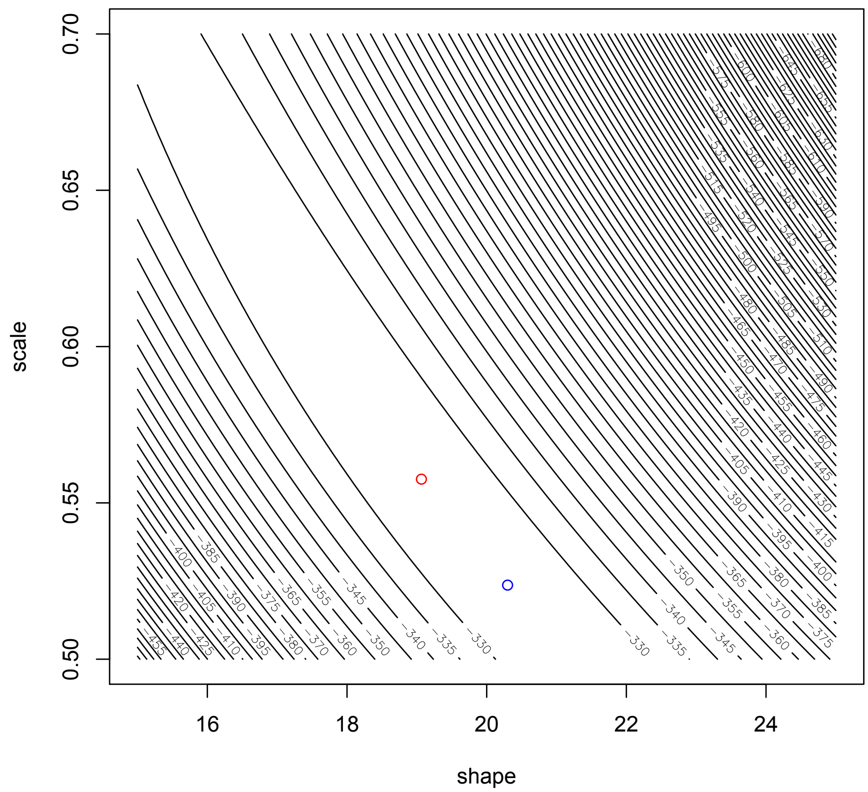
```
$argmin
     shape      scale
20.2993396  0.5236897
$final.gradient
[1]  3.864193e-05 -9.985302e-05
$iterations
[1] 5122
> cats.gamma.loglike(fit.ml$argmin)
[1] -325.5476
```

That is, the final estimate is a shape of 20.3, and a scale of 0.524 grams, with a log-likelihood of -325.55. (`argmin` is a mis-nomer here, since this is really an argmax!)

7. SOLUTION: Let's try adding the maximum likelihood estimate to the previous figure, to see where it lies in relation to the method-of-moments estimator and to the likelihood contours.

   As Figure 3 shows, the two estimates are close, and the maximum likelihood value is comfortably between the two highest plotted contours. This is as it should be.

   (The broader lesson is that when doing optimization, it is often a good idea to start one's search from either a quick but not crazy initial guess (like the method-of-moments estimate) or from a visual inspect of the function being optimized.)




```
surface.0(cats.gamma.loglike,from.x=15,to.x=25,from.y=0.5,to.y=.7,nlevels=100,
  xlab="shape",ylab="scale")
points(x=fit.mm["shape"],y=fit.mm["scale"],col="red")
points(x=fit.ml$argmin["shape"],y=fit.ml$argmin["scale"],col="blue")
```

Figure 3: As in Figure 2, but with the addition of the maximum likelihood estimate (in blue).